SIMNET M1 Abrams Main Battle Tank Simulation
Software Description and Documentation

12 February 1987

BBN Laboratories Incorporated

Contract MDA903-86-C-0309

SIMNET

DARPA
LT. COL JACK A. THORPE
(202) 694-8232, AUTOVON 224-8232
TACTICAL TECHNOLOGY OFFICE
DEFENSE ADVANCED RESEARCH PROJECT AGENCY (DARPA)
1400 WILSON BLVD., ARLINGTON, VA 22209-2308

CONSULTANT
GARY W. BLOEDORN, COL (USA RET)

PERCEPTRONICS
MR. RICHARD TAYLOR
(818) 884-3485
21122 ERWIN ST, WOODLAND HILLS, CA 91367-3713

BBN LABORATORIES INCORPORATED
DR. DUNCAN C. MILLER
(617) 497-3334
10 MOULTON STREET, CAMBRIDGE MA 02238

BBN DELTA GRAPHICS INCORPORATED
MR. MICHAEL CYRUS
(206) 746-6800
14100 S E 36th ST, BELLEVUE, WA 98006

Report No. 6323

Contract MDA903-86-C-0309

SIMNET M1 ABRAMS MAIN BATTLE TANK SIMULATION

Software Description and Documentation

James W. Chung
Alan R. Dickens
Brian P. O'Toole
Carol J. Chiang

January 1987

Accession For	
NTIS GRA&I	X
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

QUALITY INSPECTED 1

Prepared by:

BBN Laboratories Incorporated
10 Moulton Street
Cambridge, Massachusetts 02238

Prepared for:

Defense Advanced Research Projects Agency (DARPA)
Tactical Technology Office
1400 Wilson Boulevard
Arlington, VA 22209-2308

Preface

SIMNET: Advanced Technology for the Mastery of War Fighting

SIMNET is an advanced research project sponsored by the Defense Advanced Research Projects Agency (DARPA) in partnership with the United States Army. Currently in its third year, the goal of the program is to develop the technology to build a large-scale network of interactive combat simulators. This simulated battlefield will provide, for the first time, an opportunity for fully-manned platoon-, company-, and battalion-level units to fight force-on-force engagements against an opposing unit of similar composition. Furthermore, it does so in the context of a joint, combined arms environment with the complete range command and control and combat service support elements essential to actual military operations. All of the elements that can affect the outcome of a battle are represented in this engagement, with victory likely to go to that unit which is able to plan, orchestrate, and execute their combined-arms battle operations better than their opponent. Whatever the outcome, combat units will benefit from this opportunity to practice collective, combined arms, joint war fighting skills at a fraction of the cost of an equivalent exercise in the field.

While simulators to date have been shown to be effective for training specific military skills, their high costs have made it impossible to buy enough simulators to fully train the force. Further, because of the absence of a technology to link them together, they have not been a factor in collective, combined arms, joint training. SIMNET addresses both of these problems by aiming its research at three high payoff areas:

- Better and cheaper collective training for combined arms, joint war fighting skills
- A testbed for doctrine and tactics development and assessment in full combined arms joint setting
- A "simulate before you build" development model

These payoffs are achievable because of recent breakthroughs in several core technologies which been applied to the SIMNET program:

- High speed microprocessors
- Parallel and distributed multiprocessing
- Local area and long haul networking
- Hybrid depth buffer graphics
- Special effects technology
- Unique fabrication techniques

These technologies, applied in the context of "selective fidelity" and "rapid prototyping" design philosophies, have enabled SIMNET development to proceed at an unprecedented pace, resulting in the fielding of the first production units at Fort Knox, KY just three years into the development cycle.

In addition to the basic training applications, work is underway to apply SIMNET technology in the area of combat development to aid in the definition and acquisition of weapon systems. This is made possible because of the low cost of the simulators, the ease with which they can be modified, and the ability to network them to test the employment of a proposed weapon system in the tactical context in which it will be used, i.e., within the context of the combined arms setting.

Work on SIMNET is being carried out by co-contractors Bolt Beranek and Newman, Inc. (BBN) and Perceptronics, Inc. Perceptronics is responsible for training analysis, overall system specification, and the physical simulators, and BBN is responsible for the data communication and computer-based distributed simulation and the computer image generation (CIG) subsystems. The project is a total team effort.

DARPA is the DoD agency chartered with advancing the state of the art in military technology by sponsoring innovative, high risk/high payoff research and development.

from p. 14

Table of Contents

List of Figures

1. Introduction

The SIMNET M1 tank simulation is a real-time simulation of the M1 Abrams main battle tank that runs on a MASSCOMP 5600 workstation. It was developed in 1985-1986 at BBN Laboratories for use on the SIMNET M1 tank simulator. Written entirely in the C language, the simulation was designed to be portable to other 68020-based systems by minimizing the utilization of system dependent features. The simulation is clocked in real-time at 15 Hz in lockstep synchronization with the SIMNET Computer Image Generation (CIG) system manufactured by Delta Graphics, Inc.

The M1 simulation software development team consisted of the following BBN individuals: James Chung, Dave Epstein, John Morrison, Brian O'Toole, Alan Dickens, and Carol Chiang. In addition, the following BBN individuals made significant contributions to the simulation and should therefore be acknowledged: Art Pope, Dan Van Hook, Joe Marks, Jerry Burchfiel, Bruce Murray, Bruce Johnson, Maureen Saffi, and Duncan Miller. The individuals outside of BBN who made significant contributions to this effort are too numerous to name here, but nevertheless warrant our appreciation. Of these, Col. Gary Bloedorn, USA (Ret.) deserves special mention for his efforts and guidance, without which the project could not have been successful. *The dedicated efforts and hard work of everyone involved to produce this simulation merits a well-deserved commendation.*

1.1 Design Philosophy

The design of the simulation was based on a core philosophy of the following principles:

1. All models should be based on engineering first principles to the extent that they emulate as much as possible the behavior of their real-world counterparts.
2. All models should only emulate the behavior of their systems to the minimum level of detail required to achieve a level of fidelity that is perceived by the user as realistic and acceptable for selective-fidelity training.
3. Inter-model communication should be kept to a minimum and should only contain information that is actually transferred in the real world. To ensure that this information control is maintained, models should only communicate via a message-passing mechanism such as explicit external subroutine calls, rather than by access of system-wide global variables.

These principles were followed as much as possible in the design of this simulation.

1.2 Performance Validation

In order for the simulation to be effective in training tasks, it must be successful in reproducing the environment of the M1 Abrams tank at a level of fidelity that permits the crew to perform their specified tasks under the same workload and time constraints that they would experience in the actual vehicle. In addition, the vehicle simulation must be strictly constrained to the performance limitations of the M1 to avoid the negative training that would result if the crew became accustomed to a capability provided by the simulator but not by the actual vehicle (e.g. a maximum speed of 70 mph rather than 45 mph, a fuel tank that never ran dry, unlimited ammunition, or a main gun that always killed an opponent tank no matter where the round hit). Hence, achieving a high level of fidelity was given maximum importance in modeling functions of tactical significance, such as the hull and turret dynamics, controls and display sequencing and timing, and the ballistics characteristics. Performance validation thus became a critical element in the simulation development.

For the hull dynamics simulation development, the primary source of data was the TOTPERF simulation developed by the Land Systems Division of General Dynamics for the development of the actual M1. TOTPERF is not a real-time simulation; it is a very detailed Fortran-based batch-mode simulation that models steady-state and transient performance characteristics of the M1 hull for a given set of initial conditions. TOTPERF provided a baseline to which we could match the performance results of the SIMNET hull dynamics simulation.

For the controls and display simulation software, the primary documentation utilized were the M1 Operator's Manuals (TM 9-2350-255-10) published in 1981 by the Department of the Army.

The ballistics simulation was validated against the official Firing Tables for the 105mm M68 Cannon (FT-105-A-3) published in 1979 by the Department of the Army.

In addition, all of this documentation was supplemented with field experimentation at the Armor Center at Fort Knox to collect data on actual M1 vehicles and to hold discussions with Armor Center staff. It should be noted that the generous and enthusiastic cooperation and assistance provided by the Armor Center staff was an absolutely crucial element in the successful development of this simulation. The feedback provided by the Armor Center staff remains the final and most important mechanism for performance verification of the simulation.

2. Hull Simulation

2.1 Engine Dynamics

The M1 Abrams main battle tank is powered by the AGT-1500 turbine engine manufactured by AVCO Lycoming Corporation of Stamford, CT. This powerpack provides all power to both drive the vehicle and powers all transmission mounted accessories. The AGT-1500 is a free shaft power turbine engine with a two spool gasifier and a cross counterflow stationary heat exchanger.[1]

2.1.1 Torque/Speed Profile

In simulating any powerpack, the key characteristic is its steady-state torque speed profile. The basic assumption here is that the engine's dynamic behavior basically follows its steady-state engine map as long as the inputs fall within the acceptable operating schedule of the engine and do not fluctuate so rapidly in time as to cause transient effects.

For this model, the steady-state engine map was parameterized using a least-squares approximation to generate a function which produced torque as a function of engine speed and power setting based on the performance specifications of the AGT-1500 as published by AVCO. Figure 2-1 shows the manufacturer's engine map based on horsepower, and figure 2-2 shows the same data after parameterization using torque as the dependent variable. The linear regression analysis revealed that the engine's torque-speed characteristic was basically linear, and could be closely fit using the function:

$$TE = (5178.615 - 0.854588 \cdot NE) \cdot PS$$

where
TE = Engine Output Torque (ft-lbs)
NE = Engine Speed (RPM)
PS = Power Setting (0.0–1.0)

The AGT-1500 has a governor control system to limit the engine speed to between 900 and 3100 rpm. It is being simulated as a proportional gain controller that adjusts the power setting when the engine is out of range based on the speed error between the engine speed and the governor setpoint limit.

Report No. 6323 BBN Laboratories Incorporated

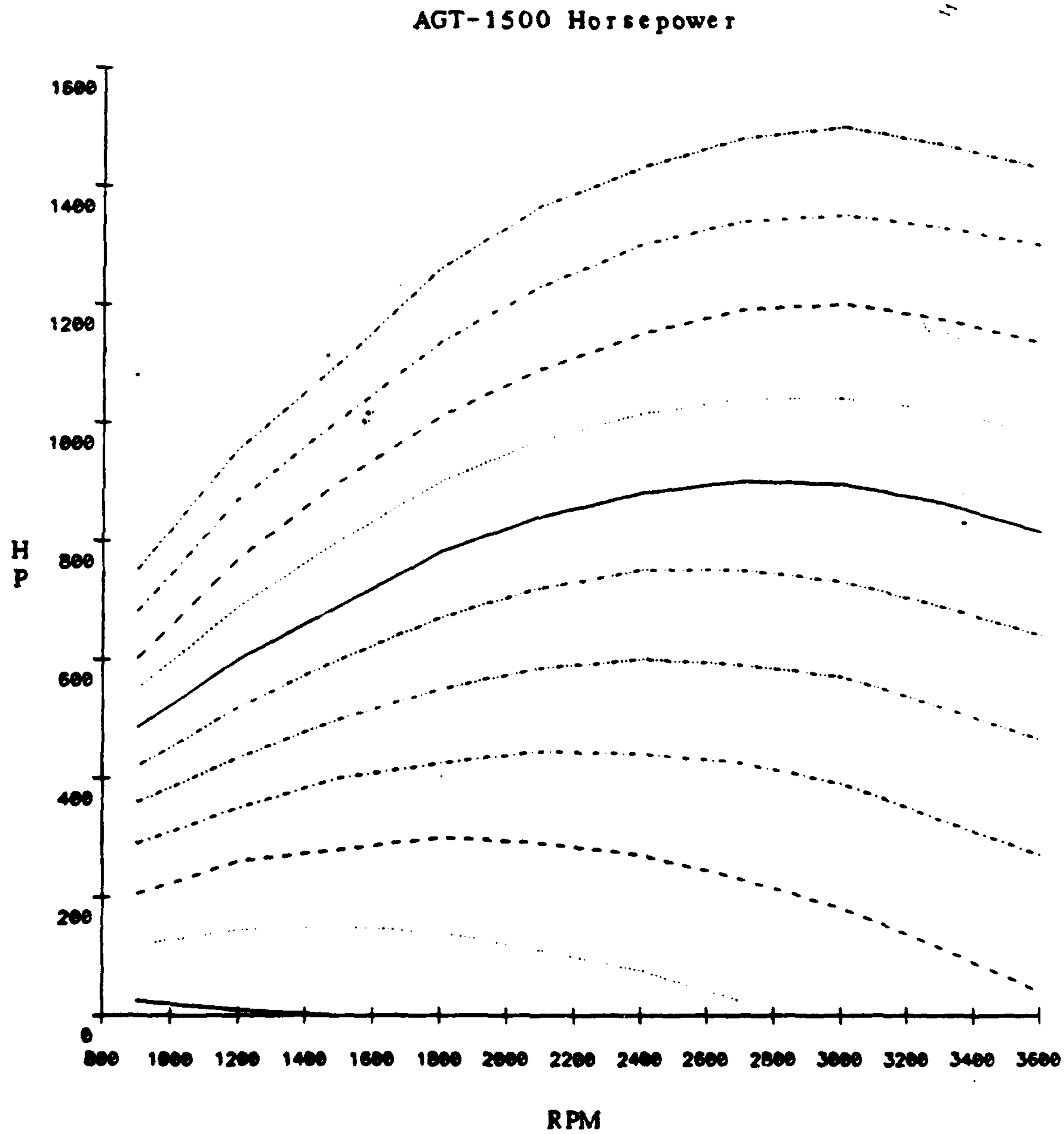


Figure 2-1: AGT-1500 Horsepower Engine Map

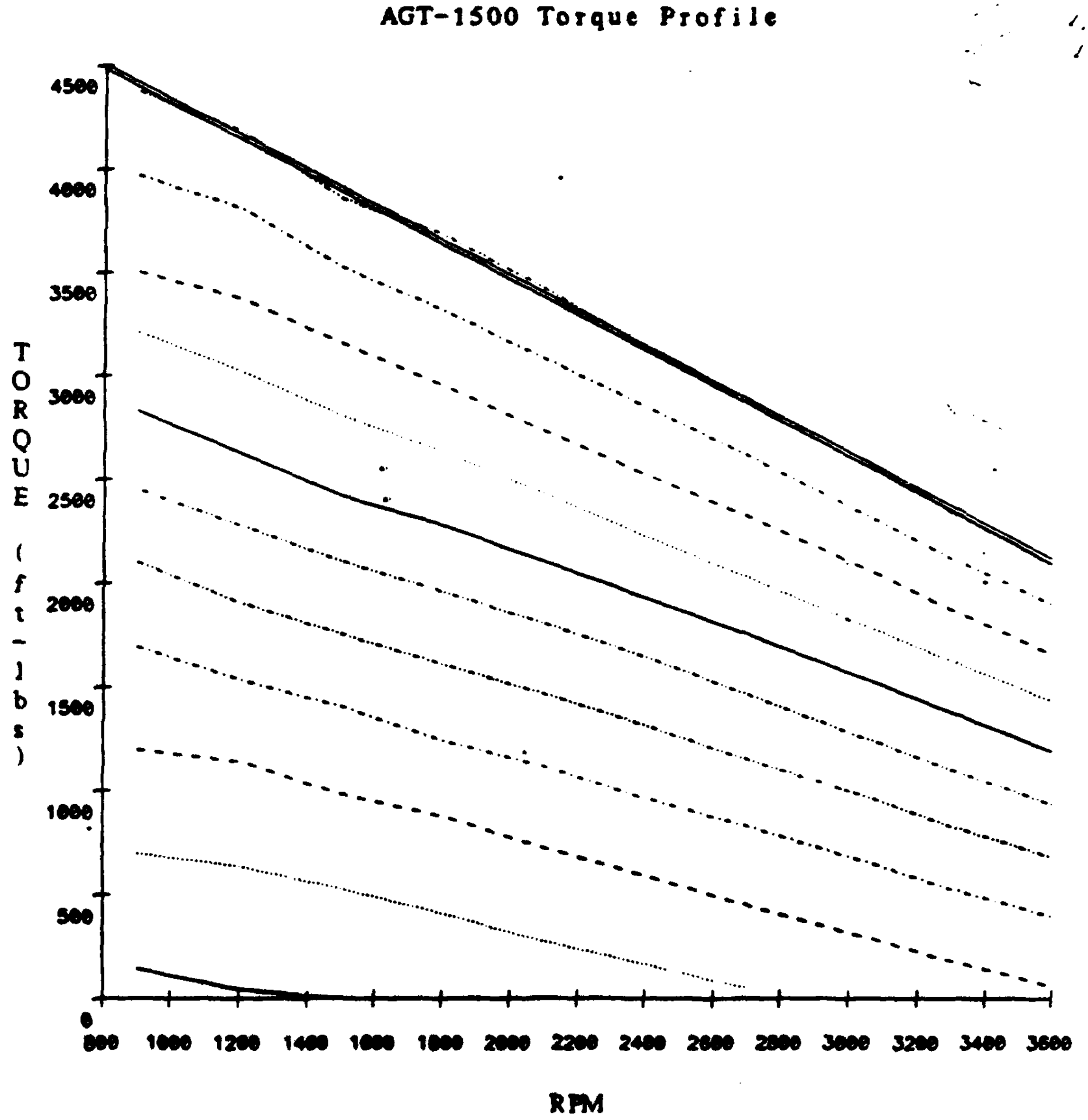


Figure 2-2: AGT-1500 Torque-Speed Engine Map

2.1.2 Fuel Consumption

The specific fuel consumption of the AGT-1500 varies to some extent as a function of where the engine is running on its performance map. These variations range from a minimum of 0.475 lbs/hp-hr at its highest efficiency to a maximum of about 0.70 lbs/hp-hr during low-speed lugging. However, since the transmission operating schedule acts to keep the engine out of its low-efficiency regions, the actual variations in specific fuel consumption during normal operation are quite small. We therefore model the specific fuel consumption of the AGT-1500 as a constant. The specific fuel consumption of the engine was empirically determined to be approximately 0.55 lbs/hp-hr by matching the simulation's steady-state fuel consumption rates with those of the actual vehicle over a wide range of operating conditions and loads.

2.2 Transmission

The Detroit Diesel/Allison X1100-3B hydrokinetic transmission is fully automatic and consists of a torque converter with an automatic lock-up clutch in combination with a four-speed planetary range gearbox.[2] In addition to transferring power from the engine to the final drive, the transmission also performs the steering and braking functions for the vehicle.

2.2.1 Torque Converter

The first stage of the transmission is the torque converter which takes the power from the engine and feeds it to the range gearbox. The torque converter also accommodates the speed differential between the engine output speed which cannot fall below 900 rpm at low idle, and the input shaft speed of the gearbox which can be as low as 0 rpm if the vehicle is stationary. The torque converter uses this speed differential to advantage by acting as a linear multiplier of the input torque for speed ratios in the range of 0 to 1. The torque converter hence is modeled with the following relationship:

$$\frac{TT}{TE} = 2.2 - 1.2\cdot\left(\frac{NT}{NE}\right)$$

where $\frac{TT}{TE}$ = ratio of TC input/output torques

$\frac{NT}{NE}$ = ratio of TC input/output speeds

From this relationship, one can see that when the vehicle is stationary, the torque converter generates its highest torque multiplication factor of 2.2. As the vehicle accelerates, the torque converter output shaft speeds up to eventually match the speed of the input shaft from the engine, during which time the multiplier drops steadily towards 1.0. When parity in the input and output speeds is achieved at approximately 1300 rpm, an automatic lock-up clutch engages which mechanically couples the two shafts together to bypass the torque converter and its associated efficiency loss. In this mode, the engine delivers power directly to the range gearbox. If, however, the vehicle slows to the point that the torque converter input shaft speed drops below 1300 rpm, the lock-up clutch disengages to prevent the engine from stalling and again enabling the torque converter function.

2.2.2 Range Gearbox

The range gearbox provides four forward and two reverse drive ratios in addition to neutral and pivot steer. The drive ratios are as follows:

FORWARD-1:	5.877:1
FORWARD-2:	3.021:1
FORWARD-3:	1.891:1
FORWARD-4:	1.278:1
REVERSE-1:	8.305:1
REVERSE-2:	2.354:1

These drive ratios are sized to distribute engine power uniformly over the vehicle speed range, maintain engine operation in the most efficient speed range, and provide the sprocket torque needed for gradeability and tractive effort requirements.[3] The transmission automatically shifts gears based on the torque converter input speed which is driven by the engine. When that speed climbs above approximately 2800 rpm, the transmission performs an upshift to the next higher gear if one is available. In FORWARD-4, since there is no higher gear available, the transmission remains in fourth and can achieve a speed higher than 2800 rpm. If the tank continues to accelerate beyond the maximum governed engine speed of 3100 rpm, the transmission can drive the engine into an overspeed condition. This will illuminate a red warning lamp on the driver's control panel.

Similarly, the transmission will automatically downshift to the next lower gear if, while the torque converter is locked up, the engine speed is forced below 1600 rpm. If no lower gear is available, the engine will not stall because the lock-up clutch will disengage allowing the torque converter to accommodate the speed differential.

The transmission select lever has five positions: DRIVE, LOW, REVERSE, NEUTRAL, and PIVOT STEER. Each of these positions enable a set of allowable gears which the range gearbox may safely enter. DRIVE, for

instance, activates gears F2, F3, and F4. To add F1 for heavy lugging or hill-climbing, the LOW position must be selected. Likewise, REVERSE activates R1 and R2, and NEUTRAL decouples the engine and torque converter from the rest of the drivetrain. PIVOT STEER is a special gear that couples the left and right output shafts to only rotate in opposite directions during a pivot steer. PIVOT STEER must only be entered when the vehicle is stationary or transmission damage will result, disabling the vehicle.

2.2.3 Steering

The differential steering system produces equal and opposite variations in transmission output speed which provide the vehicle with the ability to maintain a given turning radius in a smooth predictable manner, provided vehicle speed is not sufficiently great to cause skidout.[4] The specifications on the X1100-3B indicate the following steer ratios for each of the range gears:

FORWARD-1: 2.34:1
FORWARD-2: 1.52:1
FORWARD-3: 1.30:1
FORWARD-4: 1.19:1
REVERSE-1: 3.51:1
REVERSE-2: 1.38:1

To simulate the differential steering, an analysis was performed to determine the maximum incremental change in sprocket speed (N_S) achieved during a maximum turn at the maximum speed allowed by the transmission for each of the range gears that would result in the above steer ratios. The figures that resulted from each gear all fell very close together at about $N_S = 48.84$ rpm. Therefore, the simulation models the differential steering capability of the vehicle by scaling this figure against the steering bar angle to turn the vehicle at the appropriate steering rate. The following table indicates the steer ratios that result from $N_S = 48.84$ rpm:

GEAR	N_D	N_D+N_S	N_D-N_S	Steer Ratio
F1	122.67	171.51	73.83	2.32
F2	238.64	287.48	189.80	1.51
F3	381.24	430.08	332.40	1.29
F4	564.11	612.95	515.27	1.19
R1	86.81	135.64	37.97	3.57
R2	306.26	355.10	257.42	1.38

2.2.4 Braking

The braking system on the M1 consists of a hydraulically actuated multiple disk brake pack on each output shaft of the transmission. The simulation applies approximately 76000 ft-lbs of brake torque when the service brake is fully applied, which slows the 60 ton vehicle at a maximum deceleration rate of 14 ft/sec^2 (see figure 2-3). This is the published specification for the maximum braking capability of the M1. The matching slopes of the velocity profiles for the SIMNET M1 tank and the published specification indicate equivalent deceleration rates. (The slight offset in the profiles is due to the fact that the simulation was initialized at slightly below the nominal 45.0 mph maximum speed and therefore is insignificant.)

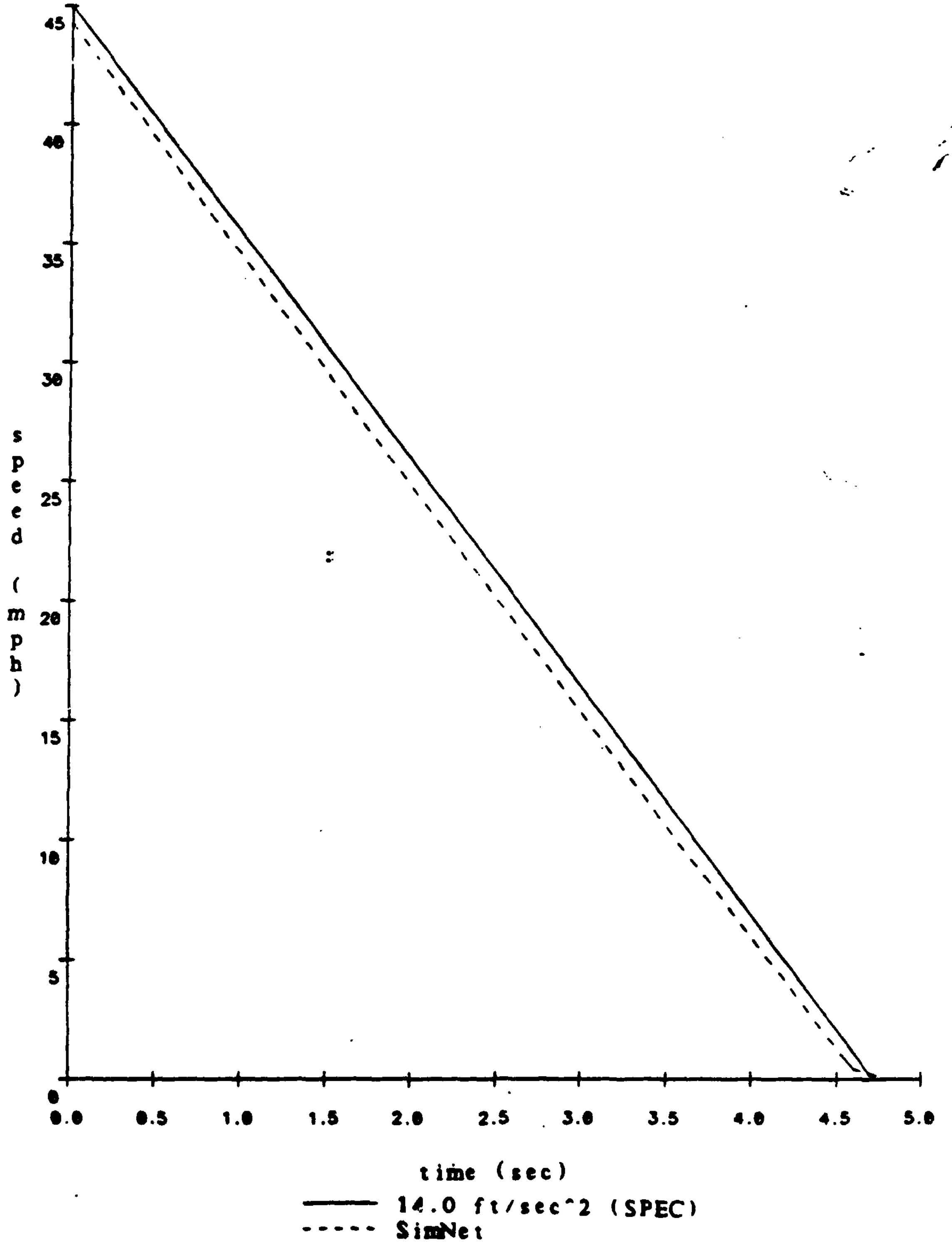


Figure 2-3: Braking Performance

2.3 Final Drive

The final drive is a coaxial planetary gear drive speed reducer which transmits power from the transmission to the sprocket hub.[5] The simulation scales all torques and speeds transmitted across the final drive in both directions by the final drive gear ratio of 4.3:1.

2.4 Drive Sprocket

The drive sprocket is the device that transforms the angular speed and torque from the final drive and delivers them to the track in the form of a linear speed and tractive force. The radius of the sprocket hub (measured to be 13.2 inches or 1.1 ft) serves as a linear scale factor in the conversion from angular to linear parameters.

2.5 Tracks

In the simulation, the tracks serve a multiple role. First, they compute the support plane on which the vehicle orients itself with respect to the terrain beneath it. Second, they interact with the soil and the earth's gravitational field to compute the loading on the drivetrain and the slippage that may result due to insufficient traction. Third, the tracks interact with the suspension system to model the dynamic effects when the vehicle encounters sudden changes in the terrain. Finally, the track model performs a collision detection check to prevent the vehicle from driving through obstacles such as houses and other vehicles in the terrain database. Each of these roles are discussed in greater detail below.

2.5.1 Support Plane Computation

The tracks compute the support plane on which the vehicle orients itself on the terrain by extracting the elevations of its three support points on the terrain polygons being sent by the SIMNET Computer Image Generation (CIG) graphics subsystem which stores all terrain information. The unit normal of this support plane determines the orientation of the vehicle as it sits on the terrain. A detailed description of the support plane computation algorithm follows in Section 2.9.

2.5.2 Soil and Gravity Interaction

The load on the drivetrain due to gravity and drag from the soil is computed each frame by the track model. The gravitational load is computed by multiplying the weight of the tank (120,000 lbs) by the sine of the pitch of the support plane. The drag force has two components: A constant force which must be overcome in order for the tank to move when it is stationary, and a component force that increases with velocity. These forces are obviously a function of the soil type. In the SIMNET terrain database, there are several different types of soils specified:

SOIL_RCI250:	Hard packed soil
SOIL_RCI050:	Soft packed soil
SOIL_WATER:	Firm riverbed (Fordable)
SOIL_MUCK:	Soft riverbed (Unfordable)
SOIL_ROAD:	Paved roadway

RCI stands for "rod-cone index", a standard by which the soil is characterized in terms of its softness and ability to support a tracked vehicle. The higher the index, the harder the soil. The simulation allows the vehicle to reach its maximum speed of 45 mph only on roads and RCI250 soil types. If the tank attempts to cross an unfordable river, it will encounter the SOIL_MUCK terrain type, which will effectively bog it down to the extent that the vehicle will be stuck and immobilized.

2.5.3 Suspension

The suspension on the M1 is a hydromechanical system which provides vehicle static and dynamic support through torsion bar springs and dampening of terrain induced vehicle oscillation by rotary hydraulic shock absorbers.[6] For the simulation, three separate suspension systems were modelled, two linear systems (one for each track) to accommodate vertical movement, and one torsional system to accommodate the pitch mode of the vehicle (see figure 2-4).

The suspension acts to provide compliance between the terrain support plane and the vehicle support plane. This permits the tank to bounce and rock accordingly when it encounters bumps and other discontinuities in the terrain, nose down during heavy braking, rear up during a sudden acceleration, and rock due to recoil effects when the main gun is fired.

The suspension is implemented with second-order filters with appropriate natural frequencies and damping ratios. The linear natural frequencies have been set to 0.56 hz with a damping ratio of 0.2. The torsional natural frequency has been set to 0.32 hz with a damping ratio of 0.3.

2.6 Drivetrain Integration Cycle

The basic drivetrain integration cycle utilizes a two-pass approach to compute the state of the drivetrain. The backward pass first computes the reflected loads at all the various stages along the drivetrain, taking into account gravity, drag, and braking loads. Once that is completed, the forward pass powers the drivetrain by taking the engine power in terms of its speed and torque and propagating it through the various multiplications and divisions of all the components discussed earlier. An integrator located at the final drive integrates the difference between the power and load torques and scales it against the reflected aggregate moment of inertia to compute the new speed of the drivetrain using the rotational equivalent of Newton's 2nd Law:

$$(T_P - T_L) = I \frac{d\omega}{dt}$$

where T_P = Power Torque
T_L = Load Torque
I = Aggregate Moment of Inertia
ω = angular shaft speed

This can be expressed alternatively as:

$$d\omega = \frac{1}{I}(T_P - T_L)\,dt$$

Integrating both sides of the equation yields:

$$\omega = \frac{1}{I}\int (T_P - T_L)\,dt$$

which is the basis of our drivetrain integration cycle.

In addition to the main integrator located at the final drive, there is a secondary integrator built into the engine simulation. This integrator becomes active only when the engine is physically uncoupled from the rest of the drivetrain (i.e. when the torque converter is active while the vehicle is stationary or moving slowly.) This secondary integrator determines the dynamic speed variations of the engine in this state independently from the rest of the drivetrain. When the lockup clutch engages to bypass the torque converter, however, the engine becomes physically locked to the rest of the drivetrain. Its speed is then determined by the main integrator in conjunction with the other components. Figure 2-5 diagrams this two-pass integration cycle for the components in the drivetrain simulation.

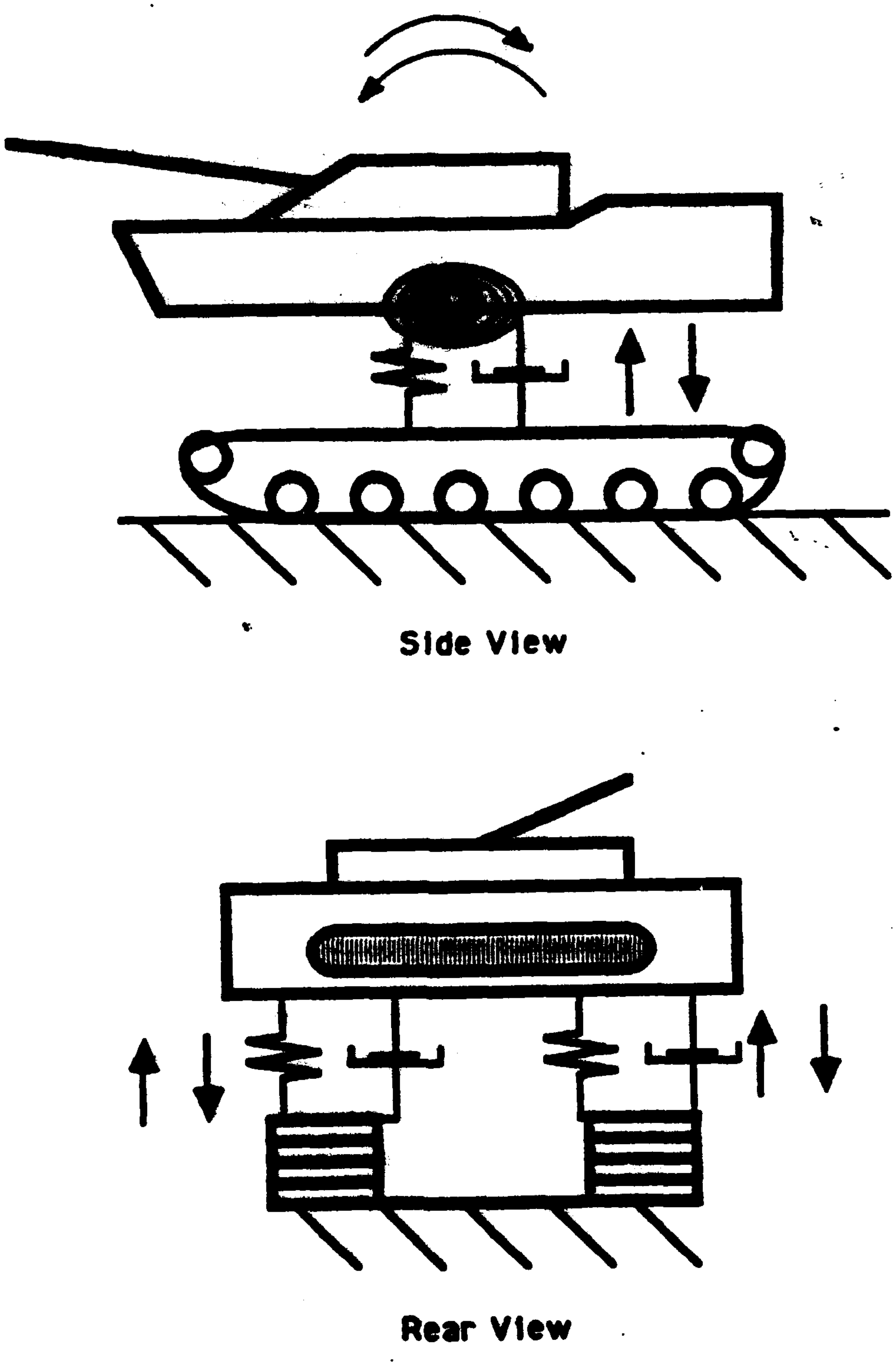


Figure 2-4: Suspension Model Schematic

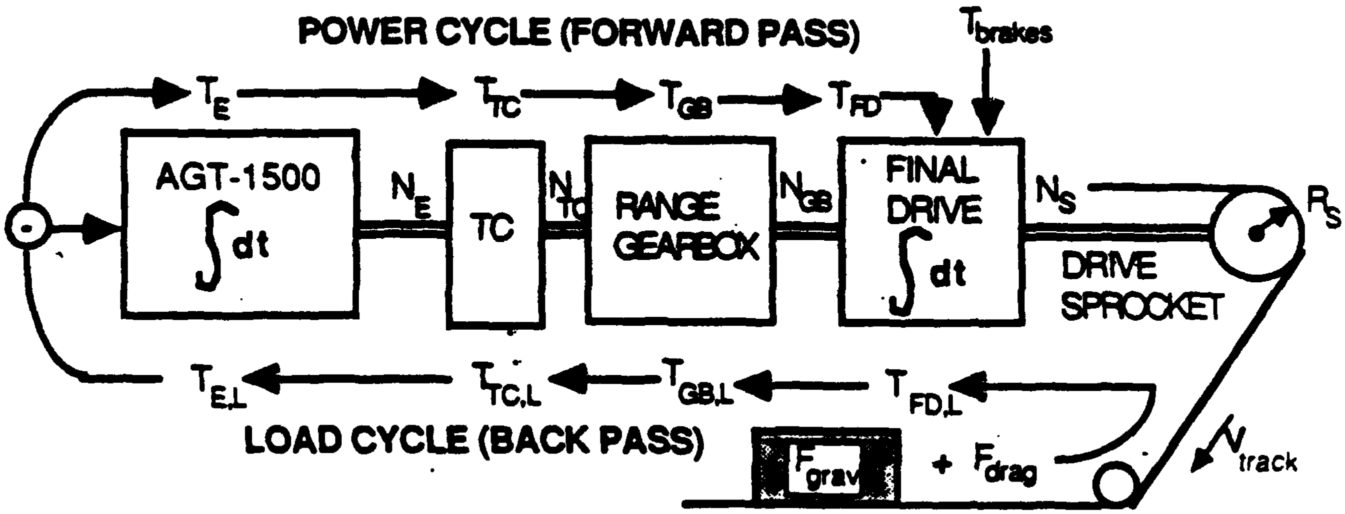


Figure 2-5: Two-pass Integration Cycle

2.7 Performance Validation

The drivetrain simulation was validated from the standpoint of both transient and steady-state performance. The results of this validation against the TOTPERF specification are presented here.

2.7.1 Transient Performance

The baseline used to verify the nominal transient performance of the model was a full-throttle acceleration from a stationary position to maximum speed. Variations on initial conditions include:

- Direction (forward, reverse)
- Soil Type (RCI250, RCI050, ROAD, WATER, MUCK)
- Grade (0%, 10%, 20%, 30%, 40%, 50%)

TOTPERF transient data for runs under these initial conditions were available to provide a baseline with which to tune the drivetrain simulation. Figures 2-6 and 2-7 display these comparison results.

2.7.2 Steady-state Performance

The steady-state validation for the drivetrain simulation focused primarily on the areas of maximum steady-state speed and fuel consumption of the vehicle at various grades ranging from -20% to 50% slopes. Figures 2-8 and 2-9 show these results along with the TOTPERF specifications.

Forward Acceleration Profile M1 Abrams (RC1-250, 0% slope)

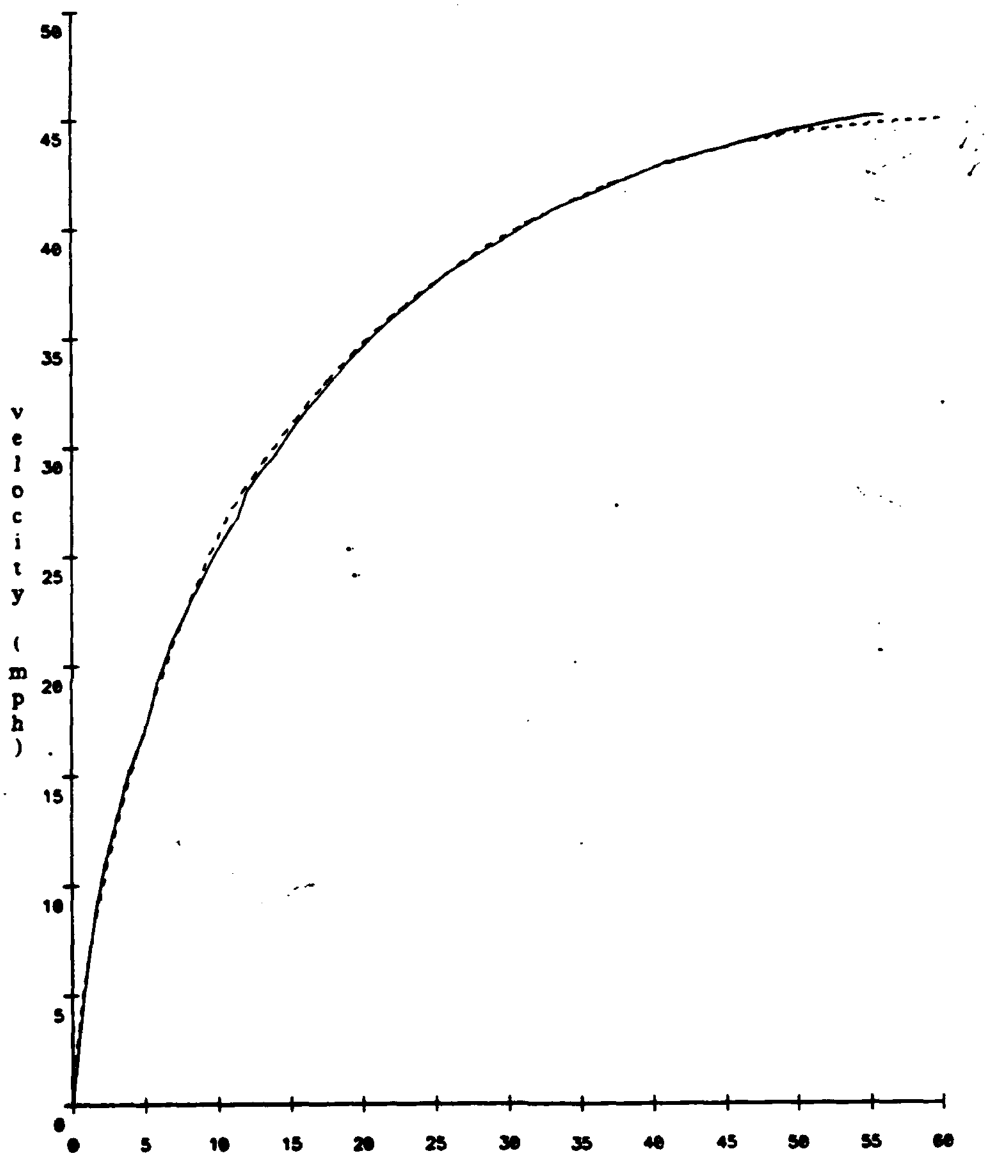


Figure 2-6: Full Throttle FWD Acceleration Profile

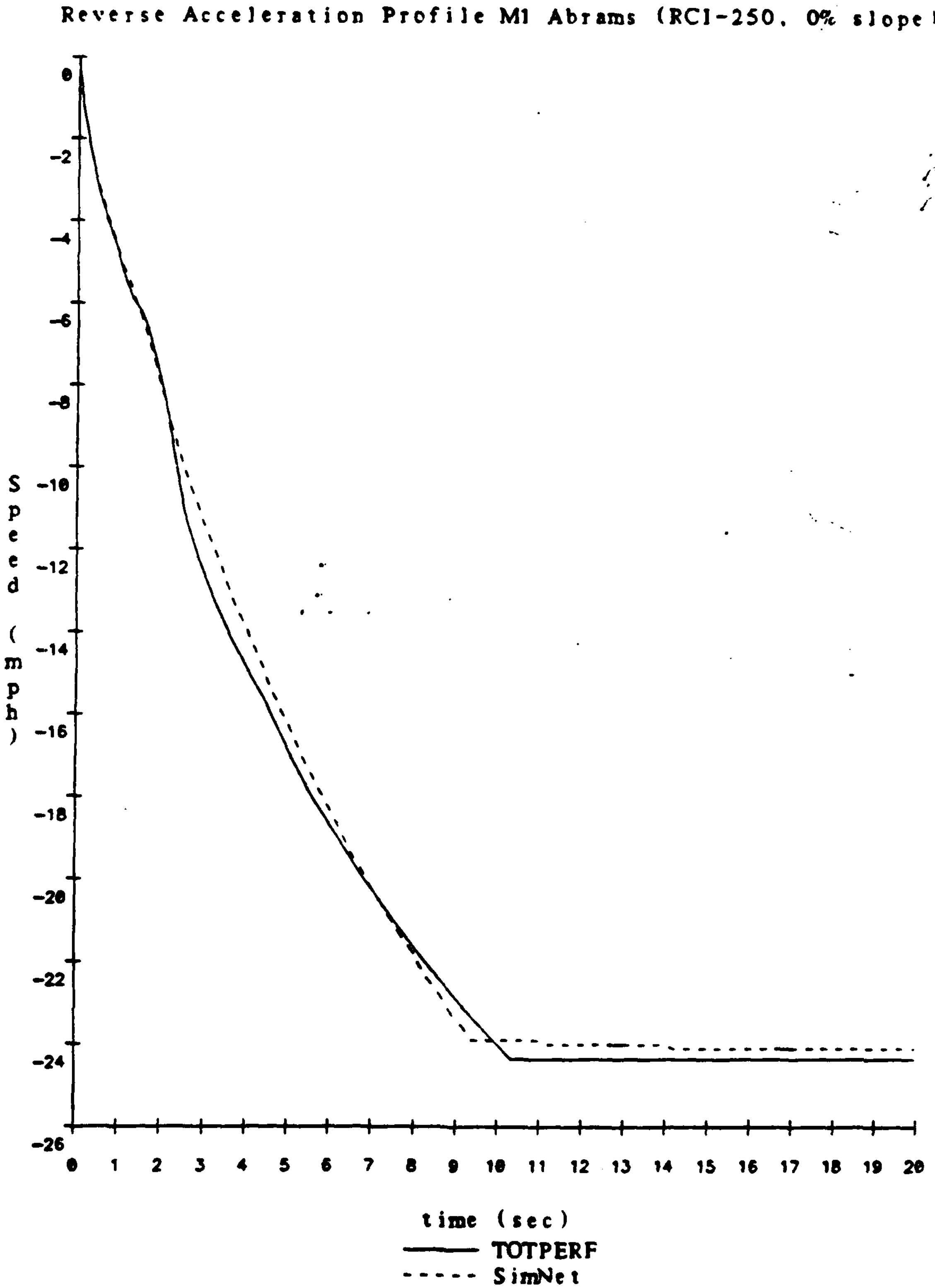


Figure 2-7: Full Throttle REV Acceleration Profile

Steady-state Hill Climbing Performance M1 Abrams (RC1-250)

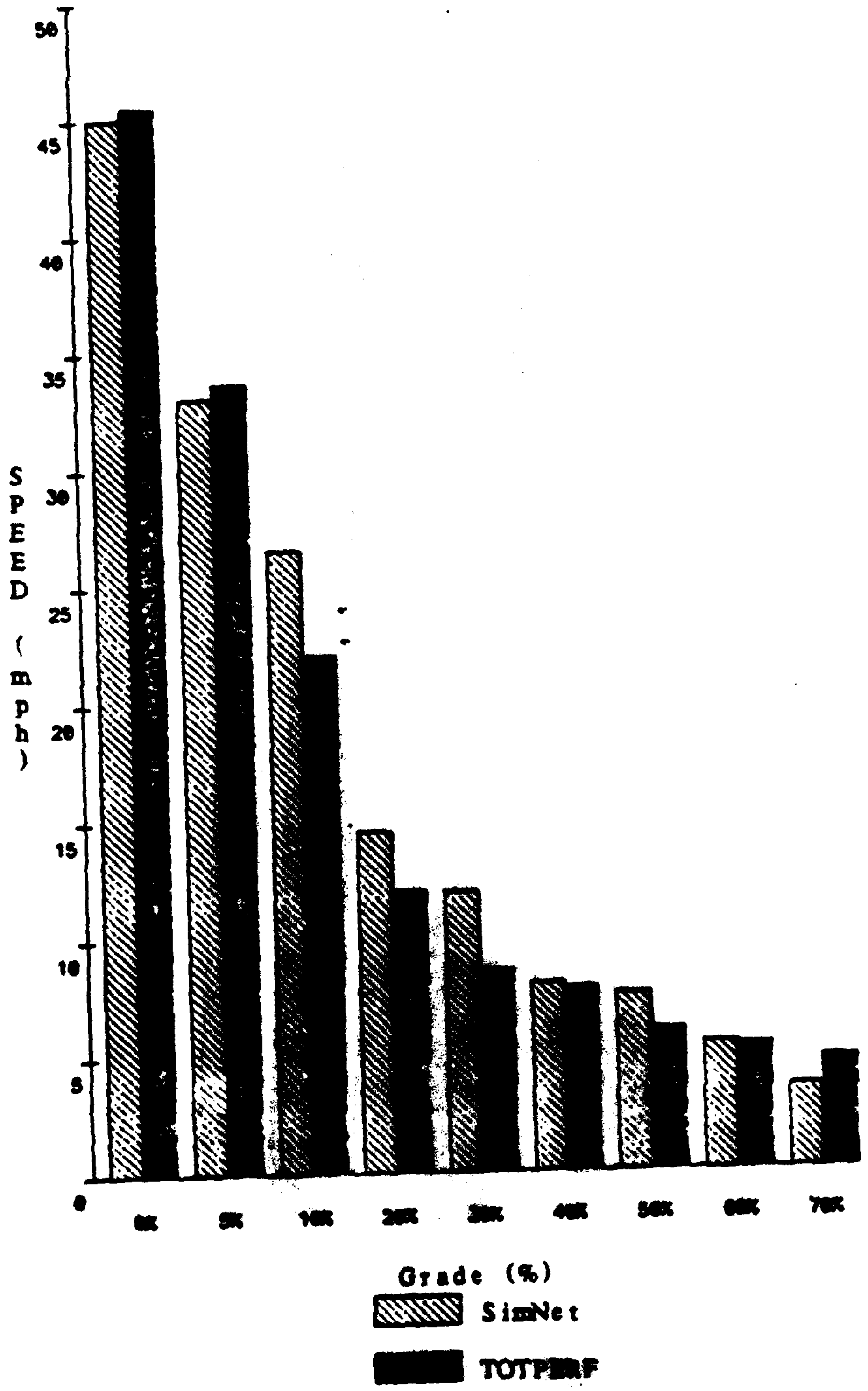


Figure 2-8: Steady-state Speed Profile

Maximum Fuel Consumption Rates, Full Throttle, RC1-250

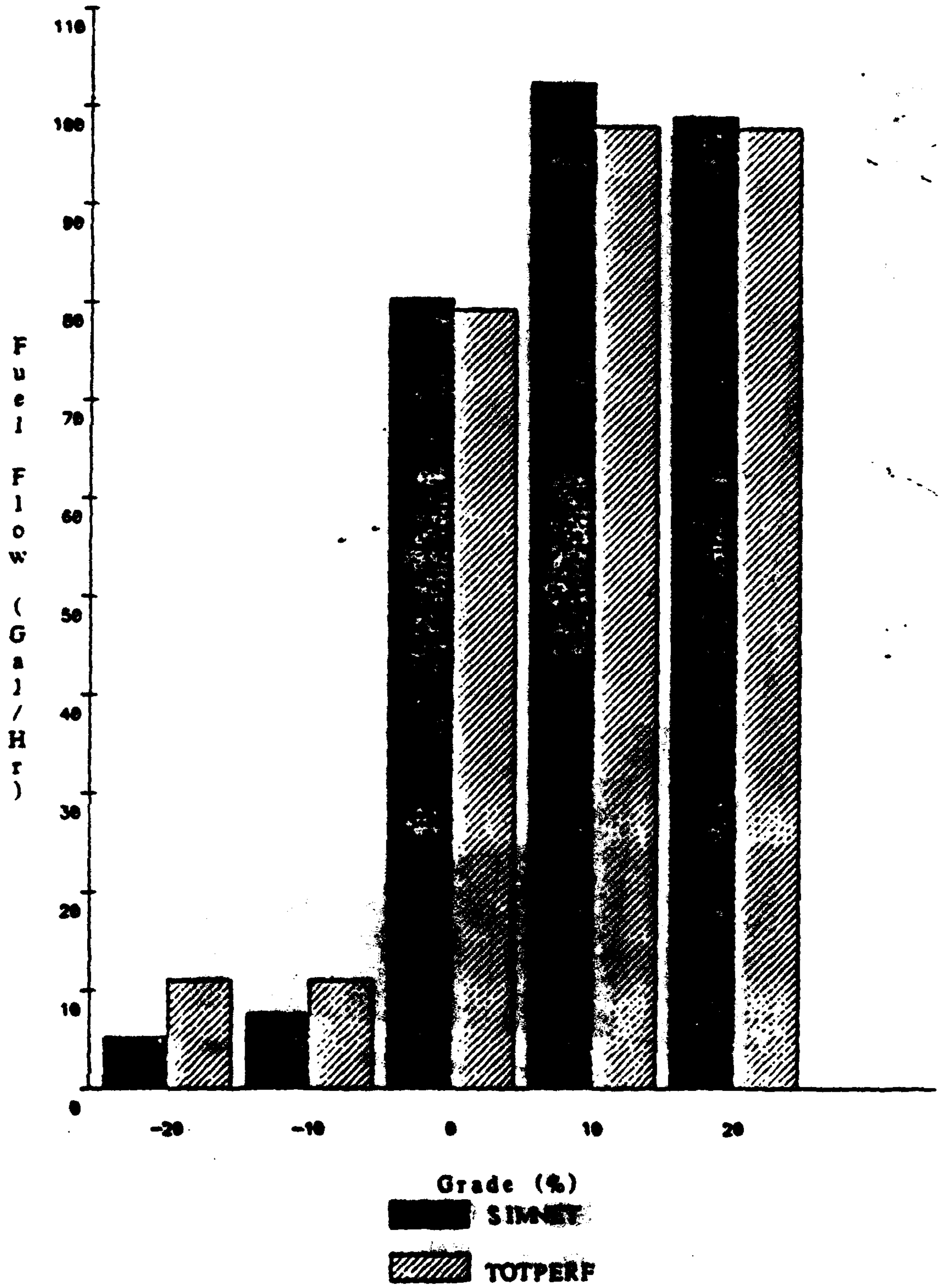


Figure 2-9: Steady-state Fuel Consumption Profile

2.8 Terrain Processing

2.8.1 Digital Terrain

The simulation receives a new "local terrain patch" (250 by 250 meters) of terrain and object polygons as well as bounding volumes from the CIG system every 37 frames (about every two and one-half seconds). The size of the patch and the update frequency are based on the maximum possible distance that the vehicle could travel in a given time. In order for the simulation to determine the position and orientation of the vehicle on the terrain, it must compute a geometric support plane from these polygons. The support plane is derived using the "bigwheel algorithm" in which the elevation of the underlying terrain is computed at the left front corner, the right front corner, and the centerline rear of the vehicle. These three points define a unique support plane that sets the orientation of the vehicle.

The procedure to determine the elevation (z) at an arbitrary (x,y) coordinate is optimized using a two-step approach. The first step is a sorting process which is performed when the local terrain patch is received. This preprocessing generates a list of polygons which might cover the point in question, greatly reducing the computation in the second step. The second step determines which polygons actually cover the point, and of these, which actually provide support for the vehicle. Unlike the first step sort which needs to be preprocessed only once for each terrain patch, the second step processing must be performed every frame.

The CIG system stores all terrain as convex and coplanar polygons. The size and orientation of the polygons can differ, although they are constrained to the following terrain patch specifications. A local terrain patch is a square 250 meters on a side. Most polygons are terrain polygons. Every vertex of a terrain polygon falls on a regular 125 meter grid, though the polygons could be either squares or triangles. In addition to terrain polygons, there are also a variety of object polygons, which represent roads, rivers, bridges, etc. These are again three or four sided, though the vertices of object polygons need not necessarily fall on the regular 125 meter grid. Finally, there are also bounding volumes (bvols), which might represent buildings, walls, telephone poles, etc. Bounding volumes are always four sided, and are represented by a polygonal footprint and a height. Like object polygons, they are not constrained to the 125 meter grid.

2.8.2 Preprocessing Digital Terrain

The preprocessing routine assumes that the local terrain patch can be divided into four imaginary "buckets", each 125 meters on a side. Each bucket contains a pointer to the polygons and bounding volumes that cover it.

Input: the number of polygons and bounding volumes, an array of polygons, and an array of bounding volumes.

Output: none.

1. Initialize the array of buckets to NULL pointers.
2. Find the lower left corner of the terrain patch. The coordinates of the bucket are based on this local "origin".
3. For each bounding volume: copy the bounding volume into a local array of bounding volumes, figure out which bucket(s) it belongs in, and place a pointer to the bounding volume in those bucket(s).
4. For each polygon: copy the polygon into a local array of polygons, figure out which bucket(s) the polygon belongs in, and place a pointer to the polygon in those bucket(s).

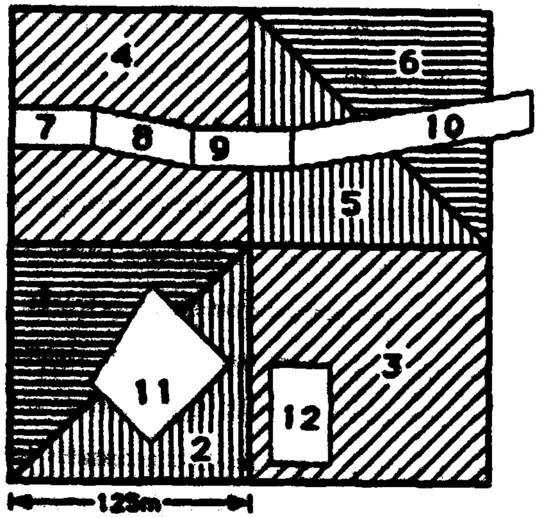


Figure 2-10: Example Terrain Patch

Consider the example terrain patch in figure 2-10: polygons 1-6 are terrain polygons, 7-10 are object polygons (perhaps a road), and 11 & 12 are bvols (perhaps buildings). In this case, the lower left bucket contains pointers to polygons 1, 2, and bvol 11; the lower right bucket contains pointers to polygons 3 and bvol 12; the upper left bucket contains pointers to polygons 4, 7, 8, and 9. The upper right bucket also contains a pointer to polygon 9, in addition to polygons 5, 6, and 10.

2.8.3 Point Inclusion Check

This algorithm is used by the elevation calculation algorithm to determine whether or not a specific polygon covers a specific point.

Input: an (x,y) location, an array of vectors which represent the edges of the polygon, an array of points which are the vertices of the polygon, and the number of vertices (and edges) in the polygon.

Output: either TRUE or FALSE - whether the point falls inside the polygon.

1. Project the point being tested, along with every vertex of the polygon into the xy plane. This is done merely by ignoring the z value (the height) of each vertex.
2. Determine a vector that represents the edge between the first and second vertices of the polygon.
3. Determine a vector that represents a segment between the first vertex of the polygon and the point being tested.
4. Find the sign of the cross product of the two vectors.
5. Repeat steps 2 through 4 for each edge of the polygon. If the signs of every cross product are the same (ignoring cross products of zero), then the point lies in the polygon. Otherwise, the point must be outside the polygon.

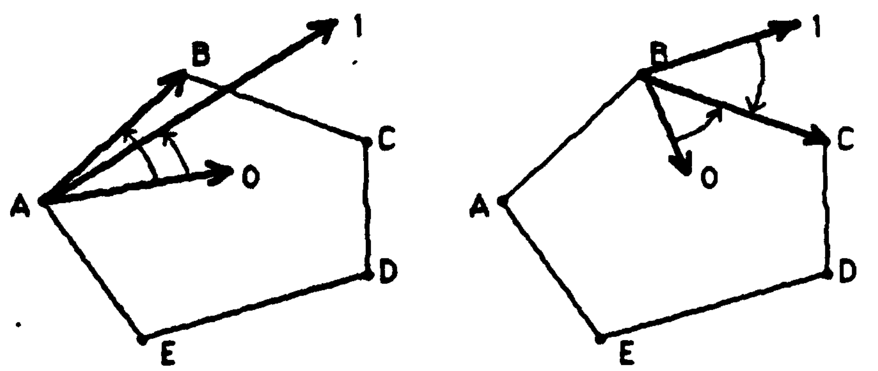


Figure 2-11: Point Inclusion Example

Consider the example in figure 2-11. This five sided polygon (with vertices A-E) is being tested against two points, 0 and 1. On the basis of the two tests shown above, point 0 seems to be in the polygon, because the cross product from both A0 to AB and from B0 to BC sweep in a counterclockwise direction. However, the same two tests conclusively prove that point 1 is not inside the polygon, because the cross product from A1 to AB sweeps in a counterclockwise direction, while the cross product from B1 to BC sweeps in a clockwise direction.

2.8.4 Calculating the Elevation at an Arbitrary (x,y) Point

The task is to compute the elevation, or Z value, at an arbitrary (x,y) location on the local terrain patch.

Input: the (x,y) location at which to find the elevation.

Output: the elevation. Note that presently, the highest elevation at the (x,y) location is returned. In the future, this routine will have to be modified to handle bridges. This will require using the z value to determine the best (closest) support polygon, not just the highest one.

1. Using the lower left corner of the local terrain patch, determine which grid bucket the (x,y) location falls in. Then, for each bounding volume and polygon that the particular bucket points to:
2. If the edge list and unit normal for the polygon or bvol have not been calculated yet, then compute them.
3. Use the edge list to check for point inclusion, by the algorithm described above.
4. If the point inclusion check is positive, then use the unit normal to compute the exact height at the (x,y) point. Given the unit normal to the polygon: $\mathbf{n} = [x_n, y_n, z_n]$, define a vector $\mathbf{p} = [x_p, y_p, z_p]$ between any *vertex* of the polygon and the (x_0, y_0, z_0) point (where z_0 is the unknown):
$$x_p = x_0 - vertex[x]$$
$$y_p = y_0 - vertex[y]$$
$$z_p = z_0 - vertex[z]$$
5. If z_p is in the polygon, then $\mathbf{n}$ and $\mathbf{p}$ must be perpendicular to each other, so $\mathbf{n}\cdot\mathbf{p}$ must equal zero:
$$\mathbf{n}\cdot\mathbf{p} = 0.0$$
$$x_n x_p + y_n y_p + z_n z_p = 0.0$$
$$z_n z_p = -(x_n x_p + y_n y_p)$$
$$z_p = \frac{-(x_n x_p + y_n y_p)}{z_n}$$
Finally,
$$z_0 = z_p + vertex[z]$$
6. Select the highest of these elevations, and return it.

2.9 Hull Kinematics

The coordinate systems used in Kinematics are:

1. World Coordinates (see figure 2-12): the Y axis points North, the X axis East, and the Z axis Up (parallel to, but in the opposite direction of gravity). The origin is at a fixed point in the terrain, currently the lower left corner of the terrain database. World coordinates are useful for describing position and orientation relative to the terrain. This is the most heavily used coordinate system.

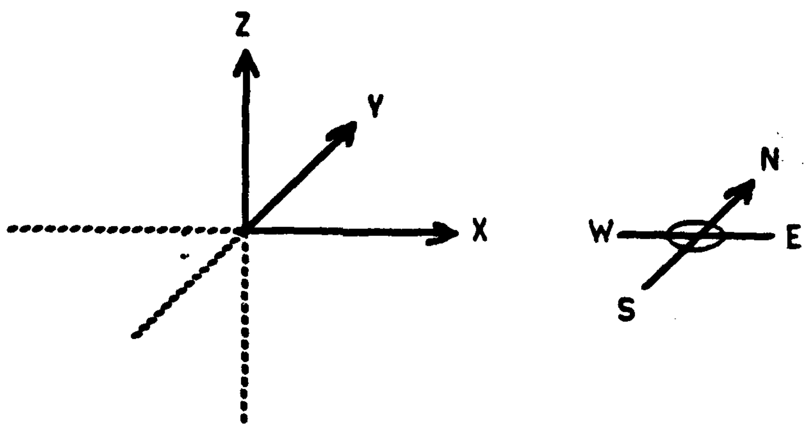


Figure 2-12: World Coordinates

2. Hull coordinates (see figure 2-13): the Y axis points through the front of the hull, the X axis through the right of the hull, and the Z axis through the top of the hull. The origin is at a point level with the bottom of the tracks in the middle of the vehicle. The Z axis vector is usually almost very close to the unit normal at the tank's position on the terrain, but it may vary slightly due to the suspension model. Hull coordinates are convenient for describing position and orientation relative to the vehicle's hull. The CIG system relies on hull coordinates as a basis for computing all display views.

Some of the key variables used in Kinematics are:

1. *world_to_hull* is a 3x3 matrix which rotates from world orientation coordinates to hull orientation coordinates. *hull_to_world* rotates from hull orientation coordinates to world orientation coordinates: it is the inverse and transpose of *world_to_hull*.
2. *hull_to_origin* is the vector from the center of the base of the vehicle to the world origin in the hull coordinate system. *origin_to_hull* is the vector from the world origin to the center of the base of the vehicle in the world coordinate system.
3. The unit normal through the top of the vehicle is stored in *unit_normal*. Note that this is identical to the third column of the *world_to_hull* matrix.
4. *delta_pos* is the positional change of the vehicle in world coordinates per frame time (DELTA_T seconds).

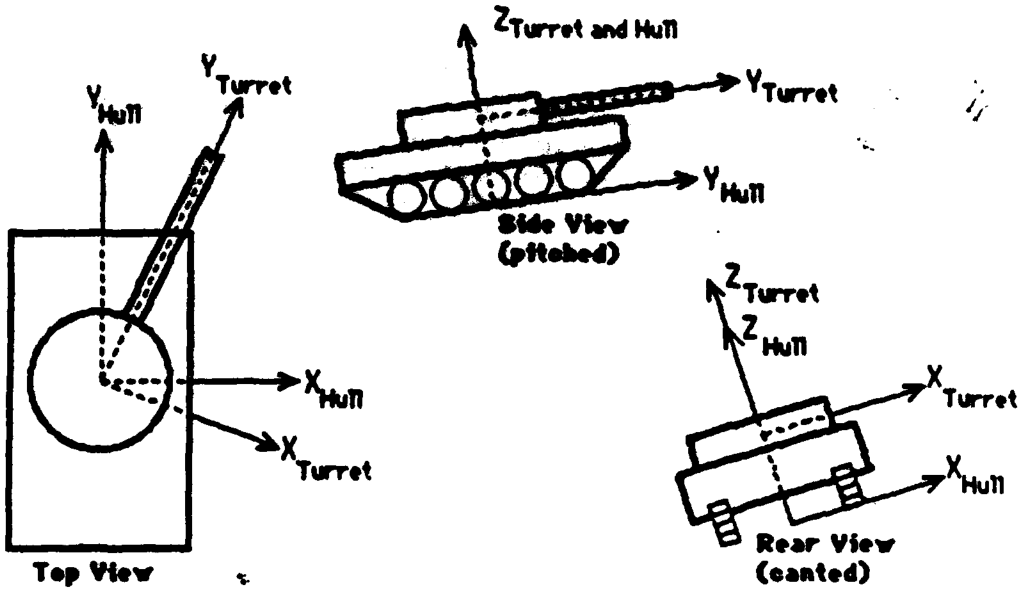


Figure 2-13: Hull and Turret Coordinates

2.9.1 Initializing a Vehicle on Terrain

Given the desired (x,y) location and heading in world coordinates, initialize all necessary variables to position the vehicle at that location with the specified orientation. This task is performed by *kinematics_init*, *kinematics_vehicle_init*, and *kinematics_simul*.

kinematics_init: initializes kinematics at the desired location and heading, but before a local terrain patch is available for support.

Input: the variables *x*, *y*, and *yaw* indicate the desired (x,y) location and heading at which to initialize the vehicle.

Output: local variables *init_x*, *init_y*, *init_yaw*, and *unit_normal* along with initial values for all variables set by *kinematics_vehicle_init*.

1. Initialize *unit_normal* to point straight up (perpendicular to the surface of the world).
2. Set up initial values for the variables which contain orientation and position. Note that this initialization does not orient the tank on any terrain, it only sets up initial position and heading.

3. Initially, set up with no collisions.

kinematics_simul: called by the main simulation loop to simulate kinematics.

1. If kinematics has been initialized with local terrain, then call the routine which updates the vehicle's orientation.
2. Otherwise (if kinematics has not been initialized with local terrain), check to see if local terrain has been initialized. If it has been, then place the vehicle on the terrain, and indicate that kinematics has been initialized with local terrain.

kinematics_vehicle_init:

Inputs: *x*, *y*, and *yaw* indicate the desired location (meters) and heading (radians) for the vehicle.

Outputs: the vehicle's location and orientation are set, utilizing the variables *world_to_hull*, *hull_to_world*, *hull_to_origin*, and *origin_to_hull*. The unit normal of the vehicle's support plane (if local terrain has been initialized) is returned in *unit_normal*, and *delta_pos* is set to all zeroes.

1. Initialize *world_to_hull* to be a rotation around the Z axis of *yaw* radians. This fixes the heading.
2. Compute the Z value, or elevation, at (x,y).
3. From the (x,y,z) location of the tank in world coordinates, compute the current value of *hull_to_origin*, which is the location of the world origin in vehicle hull coordinates.
4. Set the value of *unit_normal* by initializing the vehicle's support plane. If there is no supporting terrain, or if the terrain has not been initialized, then the unit normal does not change.
5. Set up the matrix *o_mat*, which will correctly orient the tank on the local terrain. Let u, v, and w be the first, second, and third columns of *o_mat*. To solve for the elements of this matrix:

$$\begin{vmatrix} x_0 & x_1 & x_2 \\ y_0 & y_1 & y_2 \\ z_0 & z_1 & z_2 \end{vmatrix}$$

   a. Transform unit_normal into current hull coordinates. The result is w, i.e.

$$[x_2, y_2, z_2] = [x_n, y_n, z_n]$$

   b. Since the tank is constrained along the given heading, it is constrained to lie in the yz plane in current hull coordinates. Hence *o_mat* [0][1] can be set to 0.0:

$$\begin{vmatrix} x_0 & 0.0 & x_n \\ y_0 & y_1 & y_n \\ z_0 & z_1 & z_n \end{vmatrix}$$

   c. Since *o_mat* is orthonormal, $\|v\| = 1.0$, so that

$$y_1^2 + z_1^2 = 1.0$$

   d. Also, $v \cdot w = 0.0$, so

$$x_n \cdot 0 + y_n \cdot y_1 + z_n \cdot z_1 = 0.0$$

   e. Solving c. and d. simultaneously,

$$y_1 = \frac{z_n}{(z_n^2 + y_n^2)^{1/2}}$$

$$z_1 = \frac{y_n}{(z_n^2 + y_n^2)^{1/2}}$$

f. Finally, set u = v x w, and *o_mat* has been solved.

6. Compound the heading setting transformation with the orientation setting transformation by premultiplying *o_mat* by *world_to_hull* to generate the correct value of *world_to_hull*.
7. Update *hull_to_origin* by multiplying it by *o_mat*.
8. Initialize *hull_to_world*, *origin_to_hull*, and *delta_pos*.

2.9.2 Moving the Vehicle Forward

Input: the desired distance is passed in *inc*. (If *inc* is positive, then the vehicle should be moved forward. If *inc* is negative, it should be moved backward.) The position and orientation of the tank are found in the global variables *hull_to_origin*, *origin_to_hull*, and *hull_to_world*.

Output: the new position information is stored in *hull_to_origin* and *origin_to_hull*. The positional increment in world coordinates is stored in *delta_pos*.

1. Check for collisions: if a rear collision has occurred and *inc* is negative, or if a forward collision has occurred and *inc* is positive, then tell tracks that the vehicle can't move, and return.
2. Otherwise, move the vehicle along the Y axis in its own hull coordinates by the negation of the increment. This is done by decrementing the Y component of the vector *hull_to_origin* by inc.
3. Update the values of *origin_to_hull* and *delta_pos*.

2.9.3 Turning the Vehicle

Input: the desired turning angle in radians is passed in *angle*, an angle small enough to allow the approximation $\phi = \sin(\phi)$. (If *angle* is positive, then turn the vehicle to the left: if *angle* is negative, turn to the right.) The position and orientation of the vehicle are communicated in *world_to_hull*, *hull_to_world*, *hull_to_origin*, and *origin_to_hull*.

Output: the new orientation representation is stored in *world_to_hull*, *hull_to_world*, and *hull_to_origin*.

1. Check for collisions: if a collision has occurred on the left side of the vehicle and *angle* is positive, or if a collision has occurred on the right side of the vehicle and *angle* is negative, then tell tracks to stop the drivetrain, and set *angle* to be 0.0.
2. Initialize the rotation matrix *r_mat* to be a rotation of -*angle* radians around the Z axis. (Change the

sign of *angle* because, instead of turning the vehicle, the world is turned underneath the vehicle, but in the opposite direction.)

3. Post multiply *world_to_hull* and *hull_to_origin* by *r_mat* to effect the turn.
4. Update the values of *hull_to_world* and *origin_to_hull*.

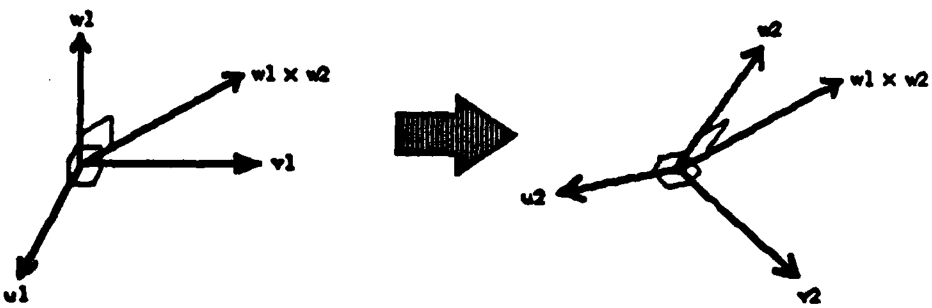
2.9.4 Computing the New Orientation after a Change in Position

As the tank moves across terrain its position and the unit normal of its supporting plane change. The task is to update all the position and orientation data structures that change due to alteration in position or the unit normal.

<u>Input</u>: current position and orientation is found in the global variables *hull_to_origin*, *origin_to_hull*, *unit_normal*, and *world_to_hull*.

<u>Output</u>: the following variables are update to reflect the new position and orientation: *hull_to_origin*, *origin_to_hull*, *world_to_hull*, *hull_to_world*, and *unit_normal*.

1. Get an updated value for the vehicle's unit normal. If there is no supporting terrain, or if the terrain has not been initialized, then the unit normal is not changed.
2. With the new value for *unit_normal*, rotate into the new support plane about an axis that is the cross product of the old and new unit normals (figure 2-14):



$$[u2, v2, w2] = [u1, v1, w1] \left| \begin{matrix} u1 \cdot u2 & u1 \cdot v2 & u1 \cdot w2 \\ v1 \cdot u2 & v1 \cdot v2 & v1 \cdot w2 \\ w1 \cdot u2 & w1 \cdot v2 & w1 \cdot w2 \end{matrix} \right|$$

Figure 2-14: Rotating into the New Support Plane

The unknown rotation matrix must accomplish these three rotations:

a. rotate about the w1 vector until -u1 coincides with w1 x w2;

b. rotate about w1 x w2, until w1 coincides with w2:

c. reverse the rotation done in step a.

Each of these rotations can be represented by a 3x3 rotation matrix. The composite matrix formed by these three matrices is the unknown matrix:

$$\begin{vmatrix} \frac{-\mathbf{v1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{-\mathbf{u1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & 0 \\ \frac{\mathbf{u1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{-\mathbf{v1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & 0 \\ 0 & 0 & 1 \end{vmatrix} \begin{vmatrix} 1 & 0 & 0 \\ 0 & \mathbf{w1}\cdot\mathbf{w2} & -\|\mathbf{w1}\times\mathbf{w2}\| \\ 0 & -\|\mathbf{w1}\times\mathbf{w2}\| & \mathbf{w1}\cdot\mathbf{w2} \end{vmatrix} \begin{vmatrix} \frac{-\mathbf{v1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{\mathbf{u1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & 0 \\ \frac{-\mathbf{u1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{-\mathbf{v1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} \frac{-\mathbf{v1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{-(\mathbf{u1}\cdot\mathbf{w2})(\mathbf{w1}\cdot\mathbf{w2})}{\|\mathbf{w1}\times\mathbf{w2}\|} & \mathbf{u1}\cdot\mathbf{w2} \\ \frac{\mathbf{u1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{-(\mathbf{v1}\cdot\mathbf{w2})(\mathbf{w1}\cdot\mathbf{w2})}{\|\mathbf{w1}\times\mathbf{w2}\|} & \mathbf{v1}\cdot\mathbf{w2} \\ 0 & \|\mathbf{w1}\times\mathbf{w2}\| & \mathbf{w1}\cdot\mathbf{w2} \end{vmatrix} \begin{vmatrix} \frac{-\mathbf{v1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{\mathbf{u1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & 0 \\ \frac{-\mathbf{u1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & \frac{-\mathbf{v1}\cdot\mathbf{w2}}{\|\mathbf{w1}\times\mathbf{w2}\|} & 0 \\ 0 & 0 & 1 \end{vmatrix}$$

$$\begin{vmatrix} \frac{(\mathbf{v1}\cdot\mathbf{w2})^2+(\mathbf{u1}\cdot\mathbf{w2})^2(\mathbf{w1}\cdot\mathbf{w2})}{\|\mathbf{w1}\times\mathbf{w2}\|^2} & \frac{(\mathbf{w1}\cdot\mathbf{w2}-1)(\mathbf{u1}\cdot\mathbf{w2})(\mathbf{v1}\cdot\mathbf{w2})}{\|\mathbf{w1}\times\mathbf{w2}\|^2} & \mathbf{u1}\cdot\mathbf{w2} \\ \frac{(\mathbf{w1}\cdot\mathbf{w2}-1)(\mathbf{u1}\cdot\mathbf{w2})(\mathbf{v1}\cdot\mathbf{w2})}{\|\mathbf{w1}\times\mathbf{w2}\|^2} & \frac{(\mathbf{u1}\cdot\mathbf{w2})^2+(\mathbf{u1}\cdot\mathbf{w2})^2(\mathbf{w1}\cdot\mathbf{w2})}{\|\mathbf{w1}\times\mathbf{w2}\|^2} & \mathbf{v1}\cdot\mathbf{w2} \\ -\mathbf{u1}\cdot\mathbf{w2} & -\mathbf{v1}\cdot\mathbf{w2} & \mathbf{w1}\cdot\mathbf{w2} \end{vmatrix}$$

Now, let $A = \mathbf{w1}\cdot\mathbf{w2}$

$B = \mathbf{u1}\cdot\mathbf{w2}$

$C = \mathbf{v1}\cdot\mathbf{w2}$

Also recall that

$$\begin{aligned} \|\mathbf{w1}\times\mathbf{w2}\|^2 &= \|\mathbf{w1}\|^2\,\|\mathbf{w2}\|^2 - (\mathbf{w1}\cdot\mathbf{w2})^2 \\ &= 1-(\mathbf{w1}\cdot\mathbf{w2})^2 \\ &= 1-A^2 \end{aligned}$$

Then, this is the "unknown matrix" which performs the desired composite rotation:

$$\begin{vmatrix} \frac{C^2+AB^2}{1-A^2} & \frac{-BC}{1+A} & B \\ \frac{-BC}{1+A} & \frac{B^2+AC^2}{1-A^2} & C \\ -B & -C & A \end{vmatrix}$$

3. New values for the *world_to_hull* and *hull_to_origin* are obtained by postmultiplying by the rotation matrix. New values for the *hull_to_world* and *origin_to_hull* are computed from the new *world_to_hull* and *hull_to_origin*.

4. Perform Z enforcement by computing the elevation at the new (x,y) location and setting *origin_to_hull*[Z] to that value. If this is not done, the tank will drift off the surface of the terrain over time.

5. Compute a new value for *hull_to_origin* from the new *origin_to_hull*.

3. Turret Simulation

The coordinate systems used in the Turret Simulation are:

1. Turret Coordinates: the Y axis points through the front of the turret, the X axis through the right of the turret, and the Z axis through the top of the turret. The Z axis of the turret coordinate system is the same as the Z axis of the hull coordinate system. The origin of turret coordinates currently differs from the origin of hull coordinates only by an offset applied to display views by the CIG system. Turret coordinates are used by the CIG system to generate out-the-window display views for the Gunner's, Commander's, and Loader's views. They are also used as an intermediate coordinate system between, for example, hull coordinates and gun coordinates.

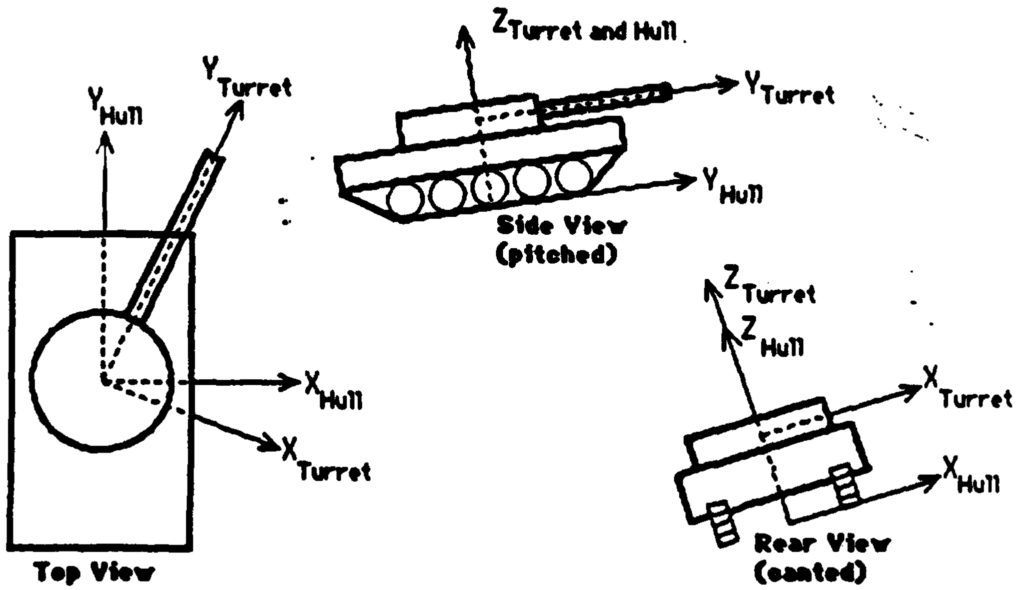


Figure 3-1: Hull and Turret Coordinates

2. Gunner's Primary Sight (GPS) Coordinates: the Y axis points in the direction of the GPS, the X axis points through the right side of the turret, and the Z axis is orthogonal to the X and Y axes. The origin is the same as that of turret coordinates. GPS coordinates are used by the CIG system to display the Gunner's view. They are also used in modelling the M1's stabilization system.
3. Laser Rangefinder (LRF) Coordinates: the LRF always points in the direction of the GPS reticle. At the present time, LRF and GPS coordinates are identical, because the system does not support a moving reticle. For a discussion of how the lead tracking system in the real M1 works, see subsection 4.3.2.
4. Gun Coordinates: The Y axis is along the gun barrel, the X axis is perpendicular to the Y axis, and is coplanar with the world xy plane, and the Z axis is orthogonal to the X and Y axes. The Y axis in this coordinate system takes into account both the superelevation of the gun, and any lead tracking that may be present. Gun coordinates are used for computing ballistic trajectories. Note that the X axis is parallel to the world xy plane to simplify the effect of gravity in the ballistic model.

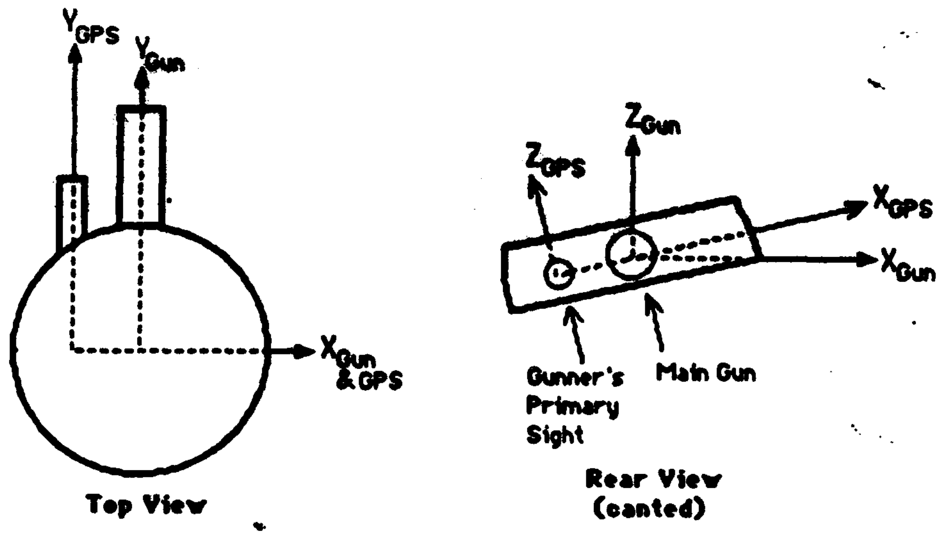


Figure 3-2: Gun and GPS Coordinates

Some of the key variables used in the Turret Simulation are:

1. *hull_to_turret* is a 3x3 rotation matrix that rotates from hull coordinates to turret coordinates - a rotation about the Z axis. *turret_to_hull* is the inverse and transpose of *hull_to_turret*.
2. *turret_to_gps* is a 3x3 rotation matrix that rotates from turret coordinates to Gunner's Primary Sight (GPS) coordinates - a rotation about the X axis. *gps_to_turret* is the inverse and transpose of *turret_to_gps*.
3. *turret_to_gun* is a 3x3 rotation matrix that rotates from turret coordinates to gun coordinates - a rotation about the X axis. *gun_to_turret* is the inverse and transpose of turret_to_gun. Note that the *turret_to_gun* matrix is usually the same as the *turret_to_gps* matrix, though under certain conditions (such as a gun elevation system failure) this isn't true.

3.1 Stabilization System

3.1.1 Setting the Target Vector for the Stabilization System

When the stabilization system on the tank is operating, it stabilizes the GPS and the main gun in both azimuth and elevation to correct for changes in hull position and orientation. One task necessary to simulate this behavior is to compute a unit vector along the axis of the GPS to act as a target vector for the stabilization system in the next frame. Note that this task must be performed even when the stabilization system is not engaged because the system may be engaged starting in the next frame.

<u>Input</u>: the *hull_to_world*, *turret_to_hull*, and *gps_to_turret* matrices are taken from global variables.

<u>Output</u>: the two vectors *gps_in_hull* and *gps_in_world*. The former represents the gps vector in current hull coordinates, and the latter represents the gps vector in world coordinates.

1. The GPS vector is trivially the vector [0,1,0] in GPS coordinates. Postmultiply this initial vector by the matrices (in order) *gps_to_turret* and *turret_to_hull*. The result is the vector *gps_in_hull*.
2. Postmultiply the *gps_in_hull* vector by the *hull_to_world* matrix. The result is the *gps_in_world* matrix.

3.1.2 Stabilization of the GPS

Given a target vector, simulation of the stabilization system requires computing the amount of rotation necessary to adjust both the turret and the GPS. These variables are then passed to the routines which slew the turret and elevate (or depress) the gun.

<u>Input</u>: the gps vectors *gps_in_world* and *gps_in_hull* are passed in global variables. The orientation of the hull is passed in the variable *world_to_hull*.

<u>Output</u>: the variables *azimuth_rot* and *elev_rot* (both radians) represent the angles necessary to rotate the turret (in azimuth) and to elevate the GPS so that the gun and GPS remain stabilized. The variable *elev_pos* (in radians) represents the angle (relative to the hull) that will keep the GPS stabilized.

1. Set two local variables: *old_gps* is the vector to stabilize to. It equals the *gps_in_world* vector multiplied by the *world_to_hull* matrix. *new_gps* is the vector to be stabilized from. It equals the *gps_in_hull* vector.
2. Imagine that the two vectors *old_gps* and *new_gps* are in a spherical coordinate system (figure 3-3).

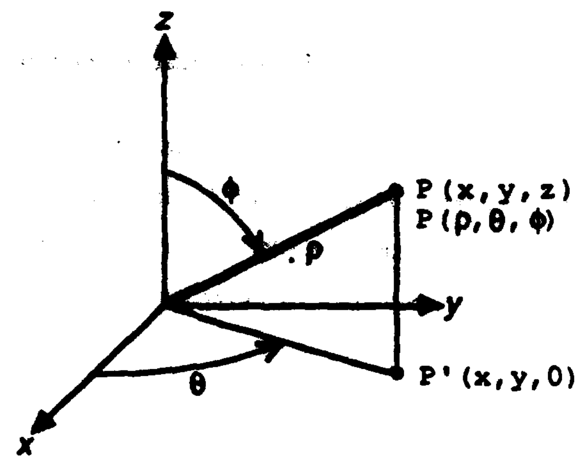


Figure 3-3: Spherical Coordinates

Recall that, for any vector $v = [v_x, v_y, v_z]$, the corresponding spherical coordinates are:

$$v_x = \rho\cos(\theta)\sin(\phi)$$

$$v_y = \rho\sin(\theta)\sin(\phi)$$

$$v_z = \rho\cos(\phi)$$

where θ is a polar angle associated with the projection of v onto the *xy* plane, ϕ is the angle between the positive *z* axis and v, and ρ is $\|v\|$.

3. Find the angle to which the GPS must be elevated to remain stabilized. This angle will be used to create the *gps_to_turret* matrix. This angle is not ϕ, but rather $(\pi/2 - \phi)$, because elevation is measured from the *xy* plane. Note, however, that

$$v_z = \rho\cos(\phi) = \rho\sin(\pi/2 - \phi)$$

and recall that the sine of an angle is the most useful piece of information from which to create a rotation matrix. Since the *old_gps* vector is a unit vector, $\rho = 1.0$, so

$$\sin(\pi/2 - \phi) = old\_gps[z]$$

will be used to create the stabilized *gps_to_turret* matrix.

4. Next, find the amount of change in elevation from the *new_gps* to the *old_gps*. The exact value for this angle is:

$$\Delta\phi = \sin[\sin^{-1}(old\_gps[z]) - \sin^{-1}(new\_gps[z])].$$

However, using small angle sine approximations, it is approximated by:

$$\Delta\phi = old\_gps[z] - new\_gps[z].$$

Since the resulting change in elevation is used only to make sure that the gun doesn't elevate or depress beyond the physical stops, an approximation is probably good enough. Note, however, that the absolute elevation of the GPS must be determined in step 3 above because the approximate change in elevation is not exact enough to keep the turret accurately stabilized.

5. Next, find the amount of change in azimuth from the *new_gps* to the *old_gps*. Consider the *old_gps* and *new_gps* vectors projected onto the *xy* plane in hull coordinates (figure 3-4).

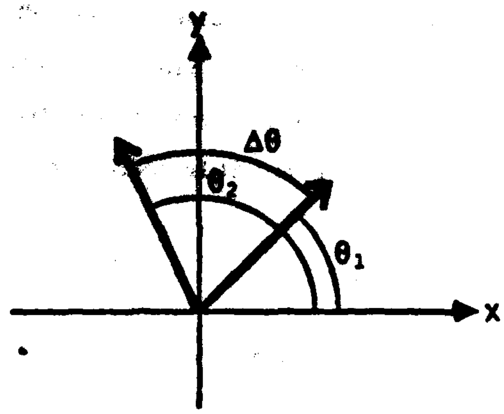


Figure 3-4: Projection of GPS vectors into hull coordinates

To determine how far to slew the turret so that the GPS remains stabilized in azimuth, Δθ must be calculated. Again, rather than knowing the value of Δθ, it would be more convenient to know the sine of Δθ (but, if at all possible, the computation of a sine function should be avoided because of time constraints).

Define two vectors **a** and **b**, such that **a** = *old_gps* and **b** = *new_gps*. First, to find the cos(Δθ), recall the law of cosines:

$$\|\mathbf{a}\|\,\|\mathbf{b}\|\cos(\Delta\theta) = \mathbf{a}\cdot\mathbf{b}$$

$$\cos(\Delta\theta) = \frac{\mathbf{a}\cdot\mathbf{b}}{(a_x^2+a_y^2)^{1/2}(b_x^2+b_y^2)^{1/2}}$$

$$\cos^2(\Delta\theta) = \frac{(\mathbf{a}\cdot\mathbf{b})^2}{(a_x^2+a_y^2)(b_x^2+b_y^2)}$$

Unfortunately, the $\|\mathbf{a}\|$ and $\|\mathbf{b}\|$ are not identically equal to 1.0, so they can't be cancelled out. However, the sine of Δθ can easily be calculated from the cosine of Δθ, and without using any trig functions:

$$\sin^2(\Delta\theta) = 1.0 - \cos^2(\Delta\theta)$$

$$\sin(\Delta\theta) = \left(1.0 - \frac{(\mathbf{a}\cdot\mathbf{b})^2}{(a_x^2+a_y^2)(b_x^2+b_y^2)}\right)^{1/2}$$

6. However, note that since the square root always returns a positive value, this equation gives the magnitude of the sine of Δθ, but not the direction of rotation. To find the direction of rotation, take the cross product: **a** x **b**. Since **a** and **b** both lie in the *xy* plane, only the *z* of the resulting vector will have a non-zero value. If the sign of this value is positive, then the cross product sweeps to the left, so the turret should be rotated in a positive direction (to the left). If the sign of this value is negative, then the cross product sweeps to the right, so the turret should be rotated in a negative direction (to the right).

3.2 Turret Kinematics

Simulation of slewing the turret and elevating (or depressing) the GPS is straightforward: the turret rotation is a rotation about the Z axis, and the elevation of the gun is a rotation about the X axis. An important point is that the turret and GPS should be moved after the stabilization corrections have been done, but before the new stabilization target vector is set. This is so that the target vector can take changes in azimuth and elevation into account.

3.2.1 Moving the Turret in Azimuth

Input: *gun_slew_handle* shows the value of the gunner's (or commander's) controls which is normalized from -1.0 to +1.0. *sin_stab_azi_rot* is the number of radians per frame that the turret must be rotated to keep the gunner's view stabilized in azimuth.

Output: a new value for the slew rate is stored in *gps_slew_rate*. The new orientation of the turret is stored in the *turret_to_hull* and *hull_to_turret* matrices.

1. The *gun_slew_rate* is passed to *turret_calc_gps_slew* to find out how far to slew the turret due to the gunner's handles. *gps_slew_rate* is set to the value returned (in radians per frame). This variable is used by the ballistic computer to determine how much lead tracking adjustment to make when the gun is fired.
2. Add the *gps_slew_rate* to the *sin_stab_azi_rot* to determine a total slew rate for the turret. Note that the small angle sine approximation is made here for *gps_slew_rate*, but not for *sin_stab_azi_rot*. Because of this, the turret will stay perfectly stabilized in azimuth as long as there is no deflection of the gunner's handles. Even when the handles are fully deflected, the maximum error in the tracking rate due to this approximation should be no more than 0.02 mils / frame, which should be within acceptable tolerances. Also note that another approximation is made by adding together an angle (*gps_slew_rate*) and the sine of an angle (*sin_stab_azi_rot*), and letting the result be a sine of an angle (*total_slew_rate*). The maximum error due to this approximation should be 0.15 mils / frame, which should not be detectable when the turret is slewing at its maximum rate of about 50 mils / frame.
3. Check the *total_slew_rate* against the maximum slew rate for the turret. If the absolute value of the *total_slew_rate* is too large, clamp it.
4. Check with the hydraulic system to make sure that sufficient pressure is available to effect the slew. If not, then set all the slewing rates to 0.0.
5. Generate a rotation matrix using the *total_slew_rate*, and postmultiply the *turret_to_hull* matrix by this matrix to effect the rotation. Also, update *hull_to_turret* to be the transpose of *turret_to_hull*.

3.2.2 Elevating and Depressing the GPS

Input: *gun_elev_handle* shows the value of the gunner's (or commander's) controls which is normalized from -1.0 to +1.0. *sin_stab_elev_rot* is the number of radians per frame that the GPS must be elevated to keep the gunner's view stabilized in elevation.

Output: a new value for the elevation rate is stored in *gps_elev_rate*. The new orientation of the GPS and gun are stored in the *turret_to_gps*, *gps_to_turret*, *turret_to_gun* and *gun_to_turret* matrices.

1. Initialize the *turret_to_gps* matrix to be the elevation specified in the global variable *sin_elev_rads*. This variable will have been set by the stabilization system to be the elevation of the GPS necessary to maintain a stabilized view.
2. Use the *gun_elev_rate* to find out how far to elevate the turret due to the gunner's handles. *gps_elev_rate* is set to the value returned (in radians per frame). This variable is then clamped so that, when added to the *sin_elev_rads*, it will not cause the GPS to be elevated (or depressed) beyond the physical stops.
3. Add the *gps_elev_rate* to the *sin_stab_elev_rot* to determine a total elevation rate for the turret. Note that the same small angle-sine approximations made in *turret_move_azimuth* are made here. Maximum errors should be smaller, because the turret can rotate at a faster rate than the GPS can elevate.
4. Check with the hydraulic system to make sure that sufficient pressure is available to effect the elevation. If not, then set all the elevation rates to 0.0.
5. Increment *sin_elev_rads* by *gps_elev_rate* to reflect the new position of the GPS.
6. Generate a rotation matrix using the *total_elev_rate*, and postmultiply the *turret_to_gps* matrix by this matrix to effect the rotation. Update *gps_to_turret* to be the transpose of *turret_to_gps*. Then, if the turret-mount interface isn't broken, copy the *turret_to_gps* and *gps_to_turret* into the *turret_to_gun* and *gun_to_turret* matrices.

3.3 Turret Dynamics

Although the actual turret drive and gun elevation systems in the M1 have some visible dynamic effects associated with them in certain circumstances, the response of the hydraulic servo drive system is sufficiently fast to enable us to model the turret via a static transfer function directly relating control handle displacement to turret slew and gun elevation rates. In making this approximation, we have encountered no complaints regarding the lack of dynamic effects in the turret simulation from any of the master gunners who have field tested the system.

3.3.1 Turret Azimuth and Gun Elevation Rate Transfer Functions

There is a relationship between the horizontal (left or right) displacement of the gunner's or commander's control handle and the resultant M1 turret slew rate. Also, there is a relationship between the vertical (up or down) displacement of the gunner's or commander's control handle and the resultant M1 gun elevation rate. A substantial amount of effort was applied to collect and analyze data on these transfer functions and to apply them to the simulation. The results of this analysis are detailed here.

The maximum turret slew rate of the M1 is 750 mils/sec, and the maximum gun elevation rate is 400 mils/sec. Positive normalized handle displacement ranges from a minimum of 0.0 (when the control handles are centered) to a maximum of 1.0. The hydraulic system transitions into a hi-speed slew mode when the normalized horizontal handle displacement is 0.33 and the slewing rate is 20 mils/sec. For gun elevation, transition occurs when the normalized vertical handle displacement is 0.6 and the elevation rate is 20 mils/sec. When the normalized handle displacement is 0.0, the slewing rate and the elevation rate should both be 0 mils/sec. Both transfer functions are odd, i.e., negative handle displacements should correspond to negative slewing and elevation rates (figure 3-5).

Displacement	Slew Rate
0.0	0 mils/sec
0.33	20 mils/sec
1.0	750 mils/sec

Displacement	Elevation Rate
0.0	0 mils/sec
0.6	20 mils/sec
1.0	400 mils/sec

Figure 3-5: Control Handle Displacement Transfer Functions

The first analysis determined the theoretical slew rates which would be required to track a moving target at various ranges and speeds. Figure 3-6 displays the results of this analysis.

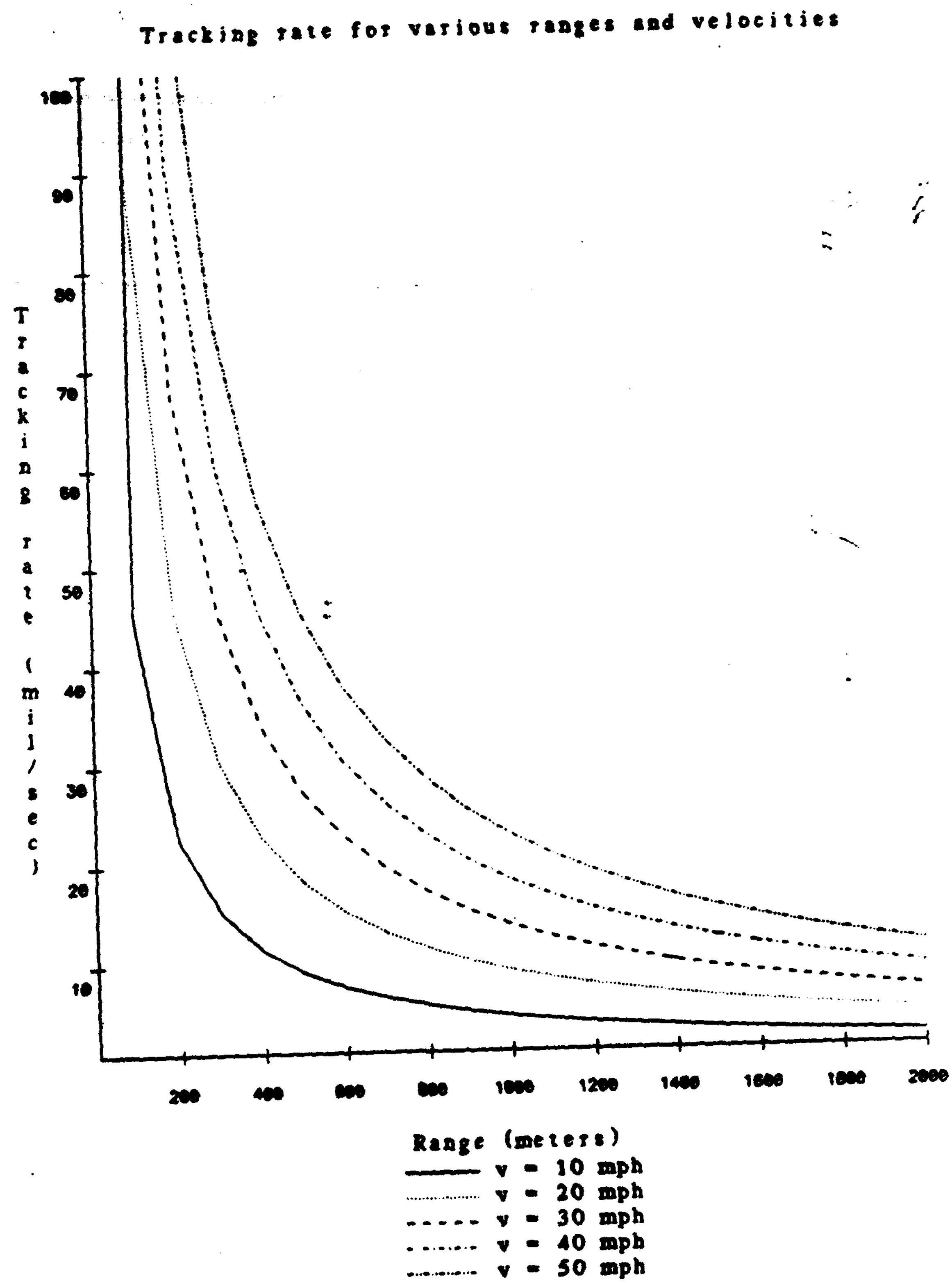


Figure 3-6: Relationship between range and tracking rate

This analysis shows that slew rates up to 40 mils/sec are necessary to track tanks moving 40 mph at a range of 500 meters. This means that any implemented transfer function must provide a level of fine control to enable a gunner to set with precision any slew rate in this tracking region of 0 to 50 mils/sec.

To provide the fine control necessary to track a moving target thousands of meters away as well as slew the turret at its maximum speed of 750 mils/sec, a continuous transfer function was implemented. This function has a low-rate fixed linear slope in the low speed regions where target tracking is performed, and a smooth transition to a curve of increasing slope to the maximum slew rate. More formally, the curve f(x) that relates normalized handle displacement to turret slew rate (mils/sec) must meet the following constraints:

1.	$f(0.33)$	$== 20$	... continuity at transition
2.	$f(1.0)$	$== 750$	... max slew rate
3.	$f'(0.33)$	$==$ slope	... continuity of slope
4.	$f''(0.33)$	$>= 0.0$	... curve turning up at transition
5.	0.33 where $f''(x) == 0$	$>= x$	... transition after point of inflection

An analysis revealed that a range of curves existed which satisfied all of these constraints. A computer program was developed which could generate any transfer function inside of this range as well as compute the upper and lower bounds of this range for a given set of input constraints. This program was used to generate the curves depicted in figures 3-7 and 3-8. These figures display the upper and lower bounds of this range, as well as the curves selected for implementation in the SIMNET M1 simulator. The curves chosen for the slew rate and elevation rate transfer functions were:

$$\omega_T = 1631.25x^3 - 1166.25x^2 + 293.75x - 8.75$$
$$\omega_E = 2916.66x^3 - 4125.0x^2 + 1833.33x - 225.0$$

where:

ω_T = turret slew rate (mils/sec)
ω_E = gun elevation rate (mils/sec)
x = normalized handle displacement (-1.0 to 1.0)

Both of these curves provide the capability for gunners to track moving targets with precision, as well as allow a smooth transition to high speed slewing and elevation to the maximum specified rates of the M1 turret and gun drive systems.

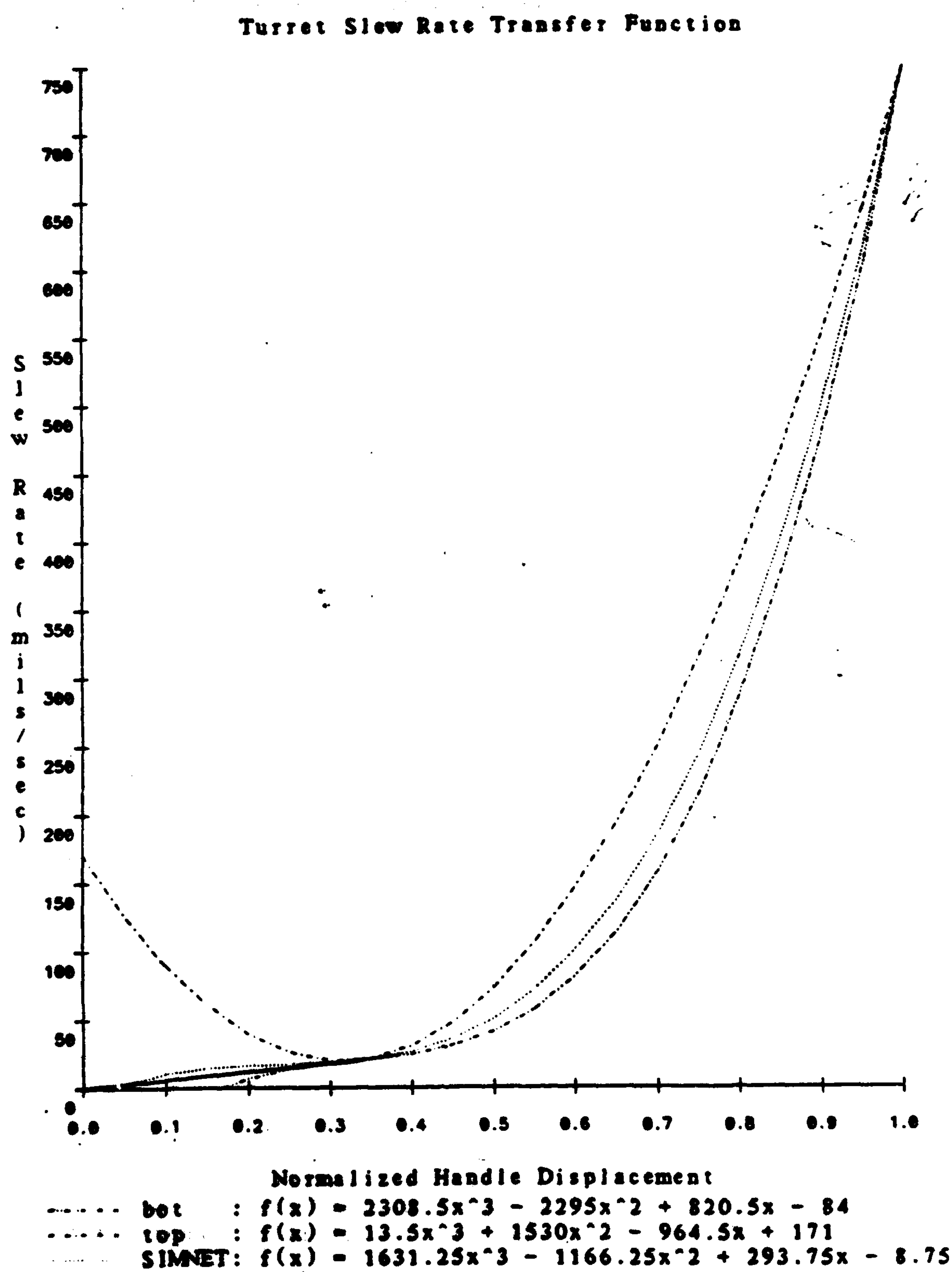


Figure 3-7: Turret Slew Rate Transfer Function

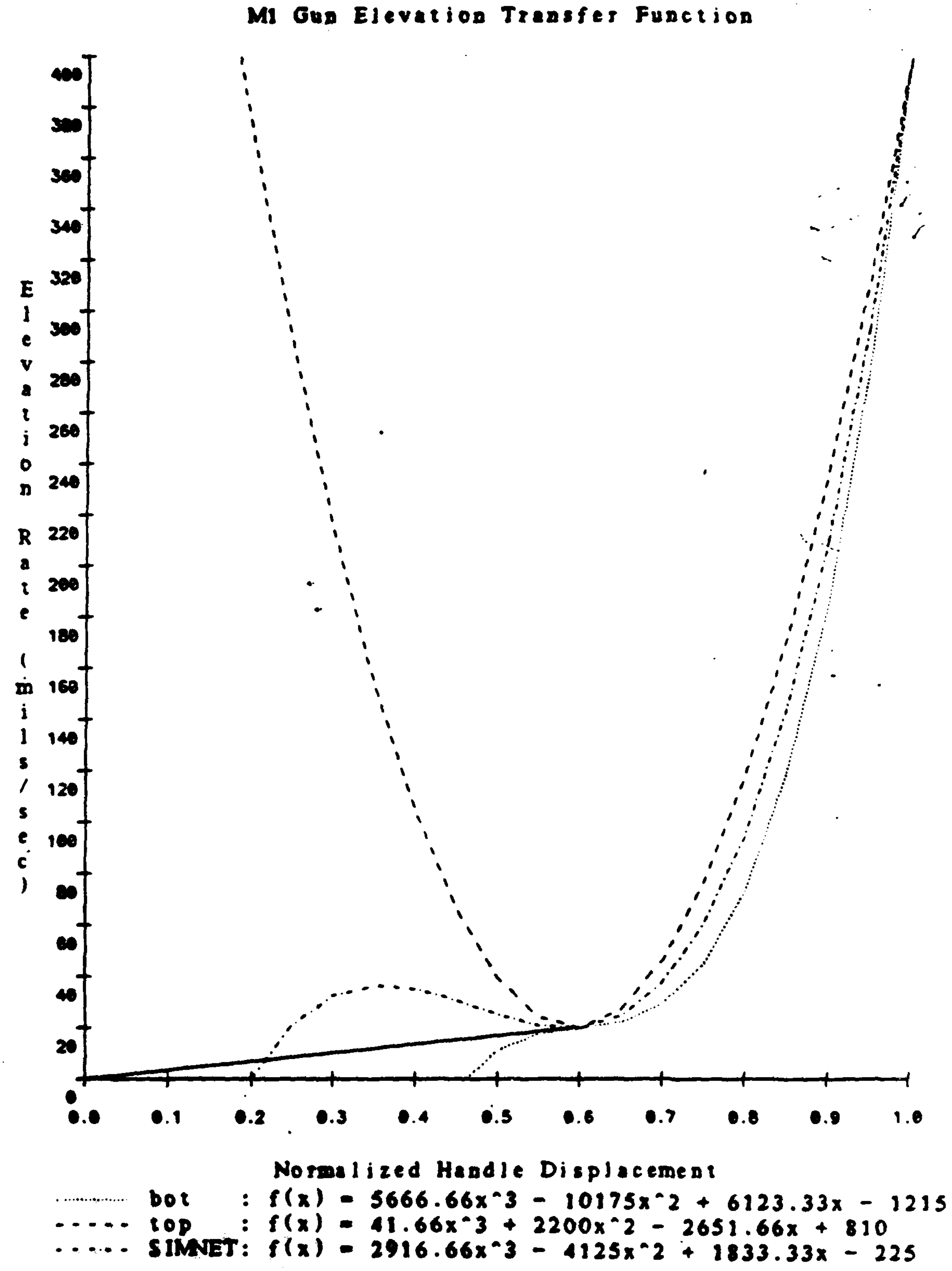


Figure 3-8: Gun Elevation Rate Transfer Function

4. Fire Control System

The SIMNET M1 fire control system consists of the components necessary for target sighting, aiming, and firing the 105mm main gun. These components include the Gunner's Primary Sight (GPS) and Commander's GPS Extension for target sighting, the Laser Rangefinder (LRF) for fast accurate ranging, and the Ballistic Computer System (BCS) for computing the ballistic solution to elevate the main gun and provide automatic lead tracking. These components in combination with the turret stabilization system provide the M1 with the capability to hit targets from both stationary positions and on the move with exceptional precision.

4.1 Gunner's Primary Sight

The Gunner's Primary Sight (GPS) and Commander's GPS extension are the primary target sighting systems for fire control in the M1. The SIMNET GPS is identical in functionality to its counterpart in the M1 with one significant exception: while the M1 GPS has a reticle projection system that is able to displace the reticle left and right from the center of the field of view, the SIMNET GPS reticle is fixed to the center. The implications of this design feature are:

1. The gunner will not observe any turret counterrotation and reticle displacement when he ranges on a moving target due to the computation of a new ballistic solution as he does in the M1. However, the system does still provide automatic lead tracking capability so he will still be able to hit moving targets with precision if he tracks the targets smoothly with the reticle and ranges accurately.
2. The gunner will not observe any reticle movement due to Gyro Reticle Compensation (GRC). The GRC system in the M1 is designed to null out any motion disturbances that are beyond the bandwidth of the turret stabilization and drive system by displacing the reticle from the center of the GPS to keep it on the target until the turret control system is able to "catch up" and zero out the azimuth error. As the error is zeroed, the reticle will relax back to the center of the GPS. In a SIMNET M1, the turret stabilization and drive system has a high enough bandwidth so that GRC is not necessary.
3. Since the reticle has no displacement capability, the gunner will not observe any visual cue to "dump lead" by releasing his palm switches after stopping from a rapid traverse. The absence of this visual cue may have a detrimental effect on the performance of some gunners who only dump lead when they see a displaced reticle. Whether or not this deficiency degrades the performance of the fire control system will be evaluated during initial tests of individual crew performance as the first set of simulators are fielded.

Maximum ordinates of M456A1 HEAT-T and M392A2 APDS-T trajectories

4.2 Laser Rangefinder

The laser rangefinder (LRF) employed in the M1 fire control system is a Neodymium Yttrium Aluminum Garget (YAG) laser transmitter coupled with a range receiver using a silicon avalanche diode detector.[7] When the unit is activated, the LRF emits a pulse and measures the time until the return signal is detected by the receiver to determine the range to the target within 10 meters. The LRF is interfaced directly to the ballistic computer so the range is automatically entered into the ballistic solution without assistance from the gunner.

In the simulation, the LRF module acquires the range to the target from the CIG front-end processor which stores the range to the nearest polygon for each pixel in its display z-buffer. The CIG interface sends the range under the center pixel in the gunsight to the simulation host every frame, although the simulation only updates the range in the ballistic solution when the laser is fired.

In the event of LRF failure or damage, the fire control system will still operate by setting the range manually. The next section discusses manual ranging in greater detail.

4.3 Ballistic Computer

The ballistic computer system (BCS) performs multiple functions for fire control. In SIMNET, the BCS simulation performs only two basic functions for fire control of the 105mm main gun. First, it elevates the gun the proper amount to send the projectile to the target range. Second, it automatically sets the proper amount of lead angle to enable the gunner to hit a moving target which he is tracking at a constant slew rate. These two basic functions of the BCS are described in greater detail below.

Another BCS capability supported by SIMNET is the ability to enter a range manually in the event of LRF failure or damage. In the actual BCS, the gunner can enter a range by the numeric keyboard on the BCS control panel. Since a SIMNET M1 simulator does not have a BCS panel, the only way to enter a BCS range manually is to depress the MANUAL RANGE BATTLESIGHT pushbutton and then incrementally add or drop to the battlesight range using the ADD/DROP toggle switch on the commander's right side panel.

4.3.1 Elevation for Range

When the gunner enters a range either manually or using the LRF, the BCS computes the elevation angle required for the round to intersect the aiming point at target range. The gun is immediately elevated by that angle over the GPS vector. This superelevation is a function not only of range, but also the ammunition type. The HEAT-T for instance, has a significantly lower muzzle velocity than the APDS-T and subsequently requires a greater superelevation to fly the same distance. One should note that the BCS computes elevation angles based on the ammunition type indexed in the gunner's control panel, NOT based on the ammunition type loaded in the breech. Therefore, if a gunner indexes an ammunition type that is not consistent with the round in the breech, the BCS will set the gun to the incorrect elevation and the round will not hit the target. This error condition can and does occur in the actual M1 fire control system and in SIMNET.

4.3.2 Automatic Lead

In the M1 tank fire control system, automatic lead tracking allows the gunner to hit a stationary or moving target by simply keeping the reticle on the target. When a stationary tank is preparing to fire on a stationary target, there is no lead, and the GPS, reticle, and gun all point in the same direction (though the gun may be elevated for range). When the tank is preparing to fire at a target which is stationary relative to the motion of the tank (even though the tank may be moving, the turret is moving only due to the stabilization system, and the gunner is not having to slew the turret to track the target), then there is no lead, and the GPS, reticle and gun all point in the same direction. When the tank is preparing to fire at a target which is moving relative to the motion of the tank (the turret is slewing because the gunner is tracking the target), then the BCS will compute a lead offset for the GPS and the reticle.

This automatic lead calculation is updated continuously as relative motion changes, and is based on the average slew rate over two seconds due to power control handle displacement. For the first 30 mils of lead offset, the reticle does not move in the GPS. Instead, the GPS rotates so that it points in a slightly different direction than the gun (and the turret). This means that whenever the gunner needs to make only minor adjustments to track the target and whenever disturbances due to tank motion are within the bandwidth that can be nulled out by the stabilization system, the reticle will stay centered in the GPS. However, the GPS has a maximum of 30 mils of freedom, so for a lead offset of greater than 30 mils, the reticle must move inside the GPS.

The current SIMNET configuration does not support this lead tracking system, because the SIMNET fire control system does not support a moving reticle. Instead, the reticle is fixed in the center of the GPS, so the simulation must support a modified lead tracking model. In this model, the GPS always points the same direction as the turret, and when fired, the gun is rotated to apply the necessary lead. Although this model does not replicate the system in the actual M1, it still allows the gunner to hit any moving or stationary target with equal precision. The lead angle θ is computed using the following relationship:

$$\theta = -\omega T$$

where θ = lead angle (rad)
ω = azimuth tracking rate (rad/sec)
T = estimated flight time to target (sec)

The estimated flight time T is a function of the ammunition type and the range to the target. Hence, in order to obtain an accurate lead angle, the gunner must have indexed the correct ammunition type on his control panel, and have a recent accurate range to the target, along with a smooth tracking rate that matches the target movement. If all of these parameters are correct and the target vehicle maintains its speed and direction during the flyout, the round has a high probability of hitting the target.

4.4 Degraded Mode Gunnery

The M1 fire control system has several levels of redundancy to enable a crew to continue an engagement in the event of a major component failure. Although SIMNET does not support the complete complement of backup systems, it does support several of the major ones to enable a crew to operate in degraded mode. These are:

- EMERGENCY MODE: In the event of stabilization system failure or other malfunction in NORMAL mode, the gunner can enter EMERGENCY mode which essentially allows him to drive the turret hydraulic drive motors directly, bypassing the stabilization system and the BCS. In EMERGENCY mode, the GPS is directly slaved to the gun, so all elevation for range and lead angle must be estimated and applied manually.
- MANUAL RANGING: In the event of LRF failure, the crew can enter the range into the BCS manually by activating BATTLESIGHT range, and then adjusting incrementally using the ADD/DROP toggle.
- COMMANDER'S OVERRIDE: In the event of gunner's control handle failure, the TC can sight, lase, and fire the main gun from his position.
- DUAL GUNNER CONTROLS: Two sets of triggers, palm switches, and laser buttons all wired in parallel are available to enable the gunner to continue operation in the event of an individual component failure or malfunction. (e.g. the left trigger)

The following degraded mode features are not available at this time:

- MANUAL MODE: Not supported due to lack of hand cranks.
- GUNNER'S AUXILIARY SIGHT (GAS): Not yet implemented. Plans call for projecting GAS range lines in GPS.
- BLASTING MACHINE: (To fire main gun without electrical power) Not supported.

5. Ballistics

The requirement for SIMNET M1 ballistics was greatly simplified in that it only required modeling the 105mm main gun for APDS-T and HEAT-T ammunition, with no requirement for the .50 caliber and 7.62mm machine guns. In addition, since SIMNET was not designed to be a gunnery conduct-of-fire trainer, several assumptions were made to greatly simplify the ballistics model. First, the main gun would be always boresighted. Second, only the nominal trajectories on an ICAO standard atmospheric day would be modelled with no crosswind conditions. [It is expected that later versions of full crew simulators based on the SIMNET R&D technology will implement a more complete ballistics model as needed for training.] The two armor defeating ammunition types modelled for SIMNET are the M392A2 APDS-T and the M456A1 HEAT-T. The SIMNET ballistics simulation is composed of the trajectory flyout computation and the hit detection computation. Each of these components is discussed in greater detail in the subsequent sections.

5.1 Trajectory Flyout Computation

After experimentation with several models to describe the aerodynamic behavior of the 105mm projectile in flight, it became clear that an aerodynamic model would be far too computationally expensive to run in real-time. The major difficulty was in developing a model to accurately describe the drag on the projectile at hypersonic velocities (often in the turbulent regime). The solution, proposed by Burchfiel et al., was to turn to a geometric rather than physical approach to modeling projectile trajectories by taking advantage of a universal "normalized" trajectory.

5.1.1 Normalized Trajectories

The basic premise of this approach is that all flying projectiles follow a "normalized" trajectory in terms of range and vertical drop angle from the boresight vector as a function of time. These normalized parameters are:

Normalized Range: $\frac{R}{R_0}$

Normalized Drop Angle: $\frac{\sin(SE)}{\sin(SE_0)}$

Normalized Time: $\frac{t}{t_0}$

Figures 5-1 and 5-2 display normalized range and drop angle as a function of normalized time. Using RS/1, a statistical analysis software package from BBN Software Products Corporation, a least-squares approximation was used to compute the coefficients for these two functions. The coefficients for the normalized range function are:

$$\frac{R}{R_0} = A5 \cdot \left(\frac{t}{t_0}\right)^5 + A4 \cdot \left(\frac{t}{t_0}\right)^4 + A3 \cdot \left(\frac{t}{t_0}\right)^3 + A2 \cdot \left(\frac{t}{t_0}\right)^2 + A1 \cdot \left(\frac{t}{t_0}\right)$$

where: $A1 = 1.0$
$A2 = -0.506721$
$A3 = 0.191974$
$A4 = -0.043222$
$A5 = 0.004268416$

The coefficients for the normalized drop angle function are:

$$\frac{\sin(SE)}{\sin(SE_0)} = B3 \cdot \left(\frac{t}{t_0}\right)^3 + B2 \cdot \left(\frac{t}{t_0}\right)^2 + B1 \cdot \left(\frac{t}{t_0}\right)$$

where: $B1 = 1.0$
$B2 = 0.148583$
$B3 = 0.016289$

The range is expressed in meters, drop angle in terms of the sine of the superelevation, and time in seconds. The normalization occurs by dividing the present range, sin(SE), and time by the "characteristic" range, sine of the superelevation, and time constants which are unique to each type of ammunition.

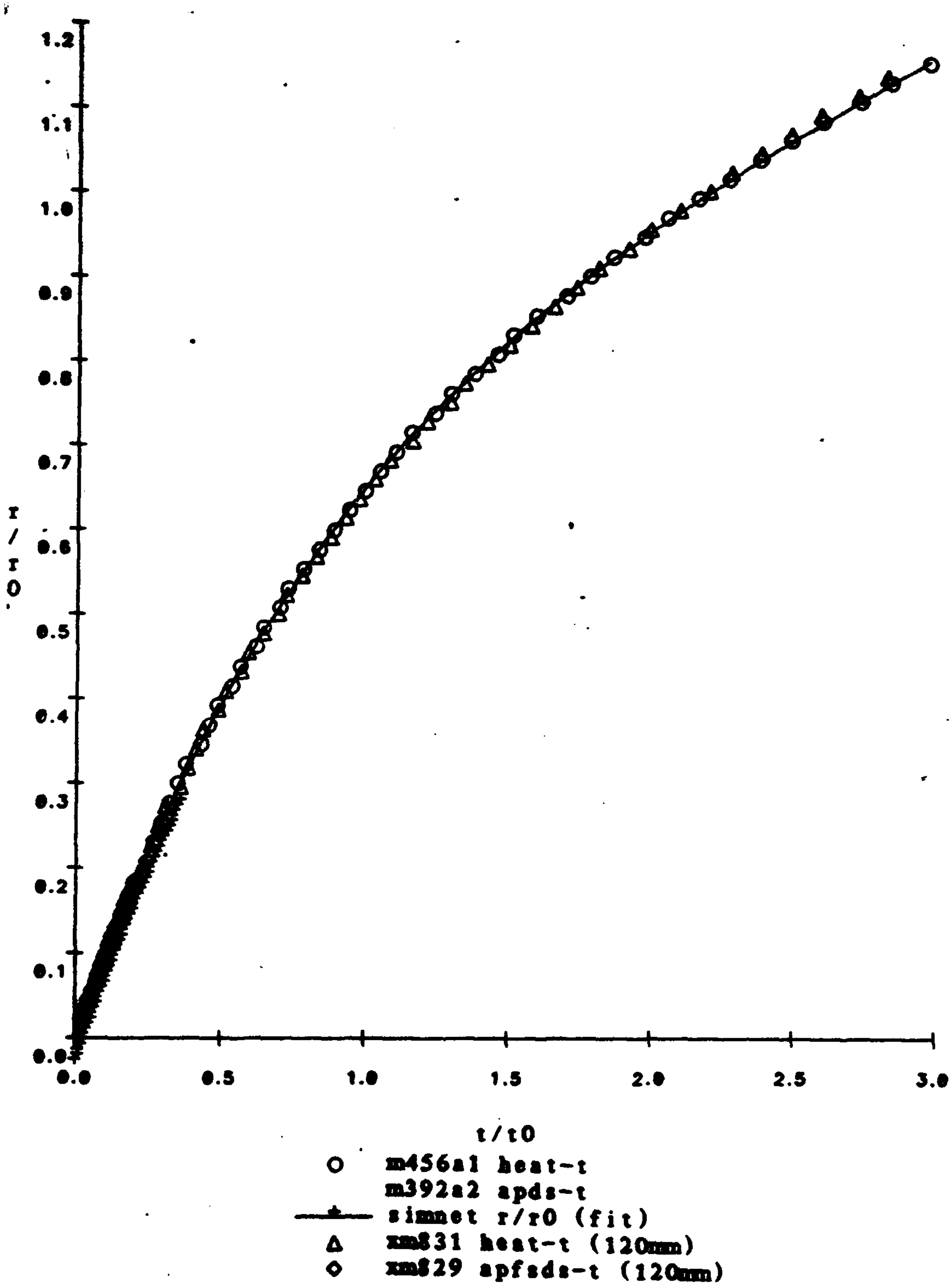


Figure 5-1: Normalized Range Function

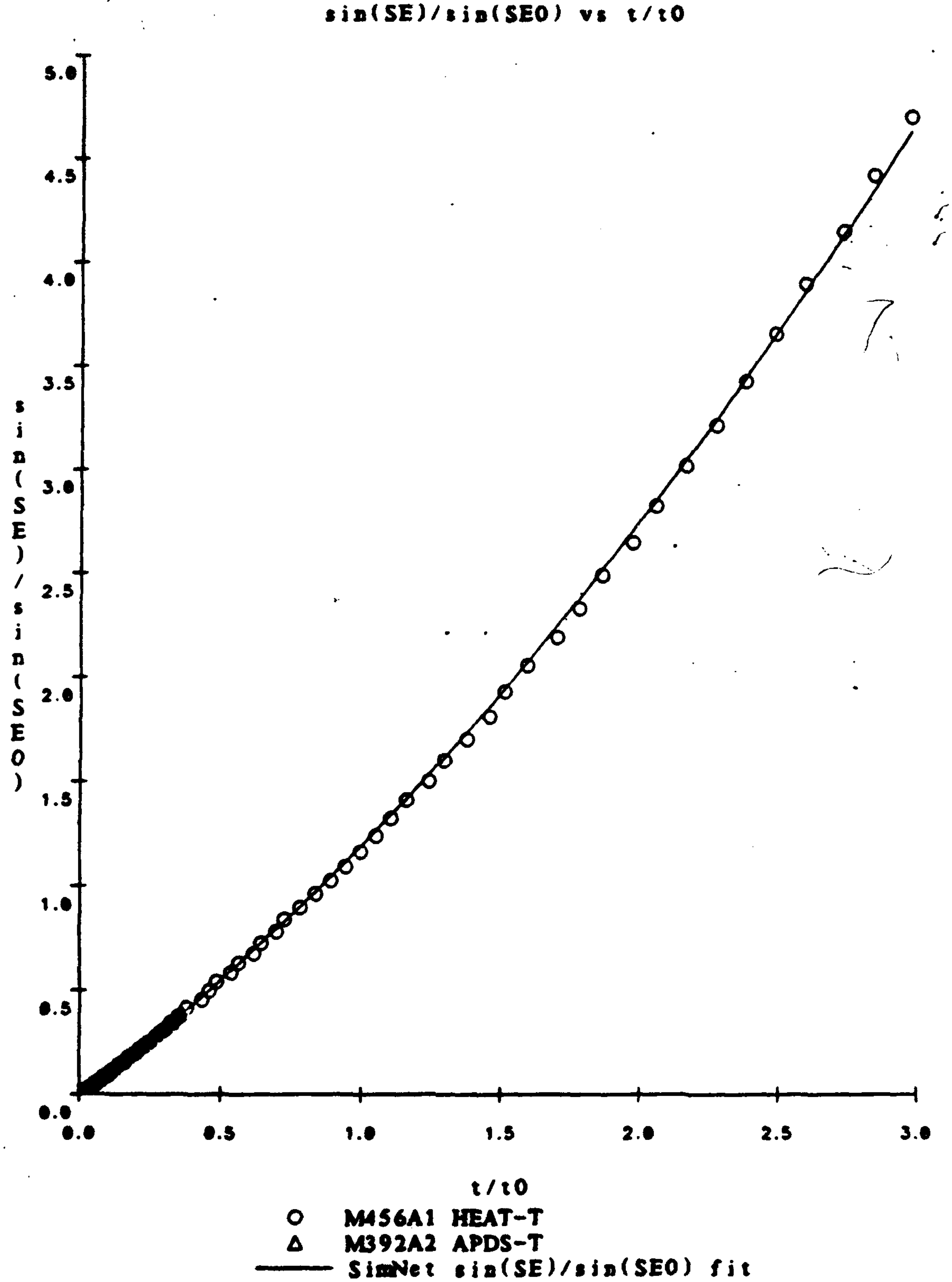


Figure 5-2: Normalized Drop Angle Function

5.1.2 Characteristic Parameters

The characteristic parameters for SIMNET projectiles are determined using the following methodology:

From the firing tables, extract the muzzle velocity (V_0) and flight time for a range of 5000 meters (t_f). Define range $R_f = 5000$ meters. Using the above approximation $\frac{R}{R_0} = f(\frac{t}{t_0})$, solve for t_0 iteratively starting from $t_0 = t_f$ such that:

$$f(\frac{t_f}{t_0}) - \frac{R_f}{V_0 t_0} = 0.0$$

Then, the characteristic range is:

$$R_0 = V_0 t_0.$$

The characteristic drop angle ($\sin(SE_0)$) is defined by the following relationship to muzzle velocity and characteristic time:

$$\sin(SE_0) = \frac{g t_0}{2V_0} \cdot (\frac{6400}{2\pi})$$

The characteristic parameters for the two 105mm rounds being modelled for SIMNET are:

Ammunition	t_0	R_0	$\sin(SE_0)$
M392A2 APDS-T	11.070	16361.6	37.38261
M456A1 HEAT-T	3.704	4344.4	15.76046

The run-time simulation works in the following manner. When the trigger is squeezed, the ballistics simulation executes a one-time procedure to compute the required superelevation. The current range in the ballistic computer (R_f) is used to compute an estimated flight time (t_f) by iteratively solving the range equation for t_f based on the input range $R = R_f$. The characteristic parameters t_0 and R_0 correspond with the ammunition type indexed by the ammunition select knob at the gunner's station. Once the estimated flight time is known, the required sine of the superelevation $\sin(SE_f)$ is easily computed by solving the drop angle equation using the characteristic parameters t_0 and $\sin(SE_0)$ for that indexed round of ammunition, and flight time $t = t_f$. This is the sine of the superelevation angle. The ballistics simulation then enters its normal flyout cycle in which it uses the elapsed flight time of the round to compute the present range $R(t)$ and drop angle $\sin(SE(t))$. It then creates a position vector B in boresight coordinates with the following components:

$$B_x = 0,$$
$$B_y = R(t)\cos(SE(t))$$

$$B_z = R(t)\sin(SE(t))$$

Finally, it transforms this position from boresight coordinates to world coordinates by multiplying this vector by the *gun_to_gps* matrix (a rotation matrix initialized to rotate the gun up from the GPS along the x-axis of the gun mount by the superelevation angle), then adding a translation vector of the tank's inertial reference frame coordinates based on its position and velocity when the gun was fired. At the end of each frame, a piecewise linear chord segment is generated from these coordinates and the coordinates of the previous frame, and passed to the CIG system for tracer display and hit detection.

5.2 Error Budget Analysis

When a round is fired at a target, the first round hit probability (p_{H1}) is affected by a number of factors. The errors that contribute to ballistic dispersion include fixed biases that remain constant for every round fired (e.g. range estimation, jump, parallax), variable biases that can change with conditions (e.g. cant, crosswind, ambient temperature, air density), and random biases that will be different for every round fired (e.g. laying error, round-to-round variations). Some of these biases are bi-variant, and others occur in one axis only. Since the ballistics simulation for SIMNET computes an exact trajectory for each round, it does not explicitly compute a p_{H1}. Since SIMNET only models the nominal flyouts on a standard atmospheric day with no crosswind, temperature or air density variations, many of these biases are zero. Aside from human errors (e.g. laying, ranging, jump, etc), the only error that can can affect a flyout in SIMNET is a random bias consisting of a bivariant normal distribution with a standard deviation of $\sigma = 0.3$ mils. This bias is designed to account for round-to-round variations and should produce realistic shot dispersion patterns at target range.

5.3 Hit Detection

The CIG system checks every chord segment it receives for intersection against all the polygons in the display. The CIG returns the coordinates of the the first intersection point it discovers, along with information about what the polygon is (e.g. a terrain patch, a tank turret, a tank hull, etc), and an angle of incidence. If the hit is against another vehicle, the simulator needs to pass this infomation to the target vehicle simulator so that damage assessment can be performed. For more information on combat damage modeling, see section 10.1.

5.4 Performance Validation

The ballistics simulation can be validated by simply comparing the flyouts of the simulated rounds in terms of superelevation, flight time, and maximum ordinate with the numbers specified for the actual rounds from the official firing tables (figures 5-6 and 5-7). The results of this comparison are presented in figures 5-3, 5-4, and 5-5.

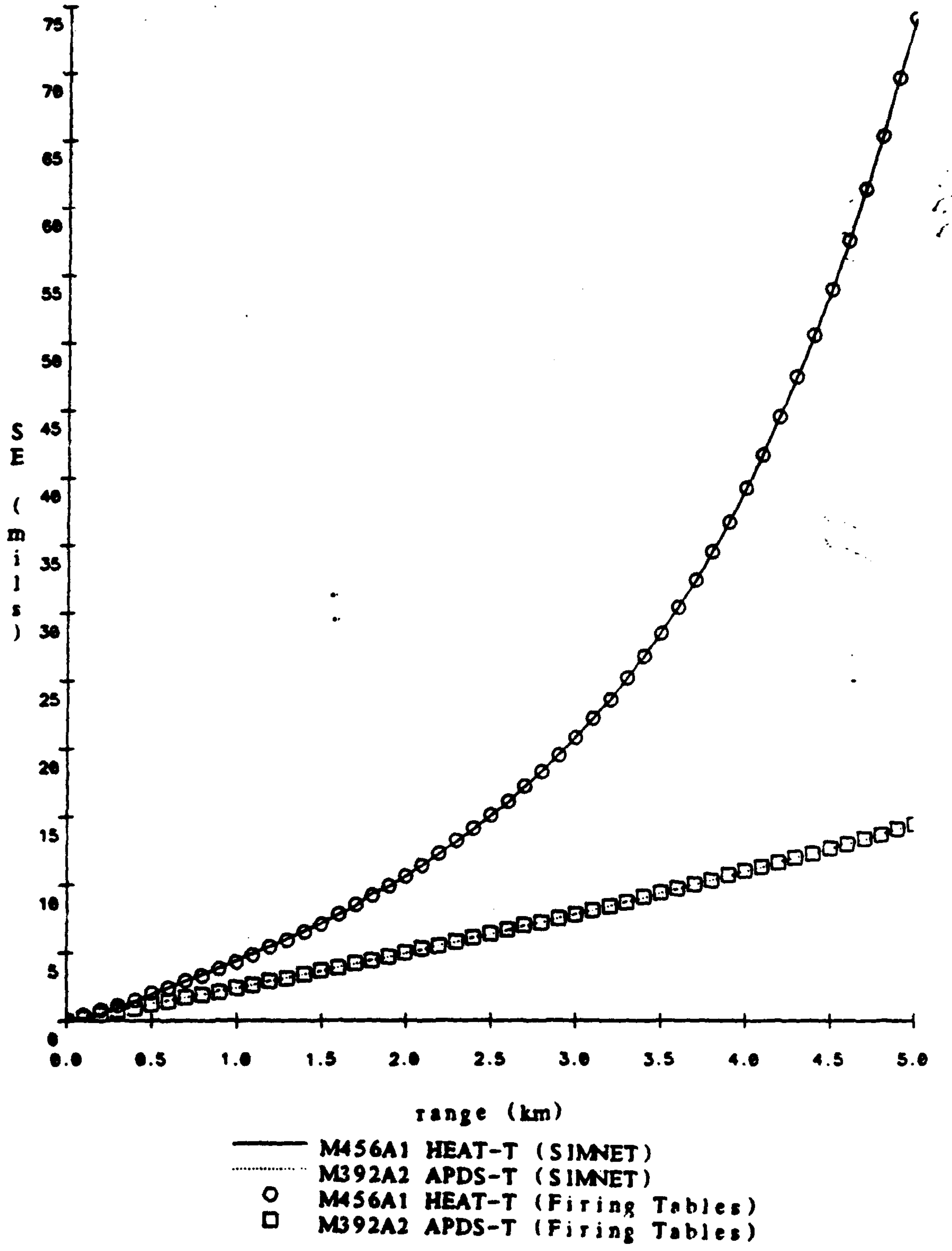


Figure 5-3: Ballistic Validation - Superelevation

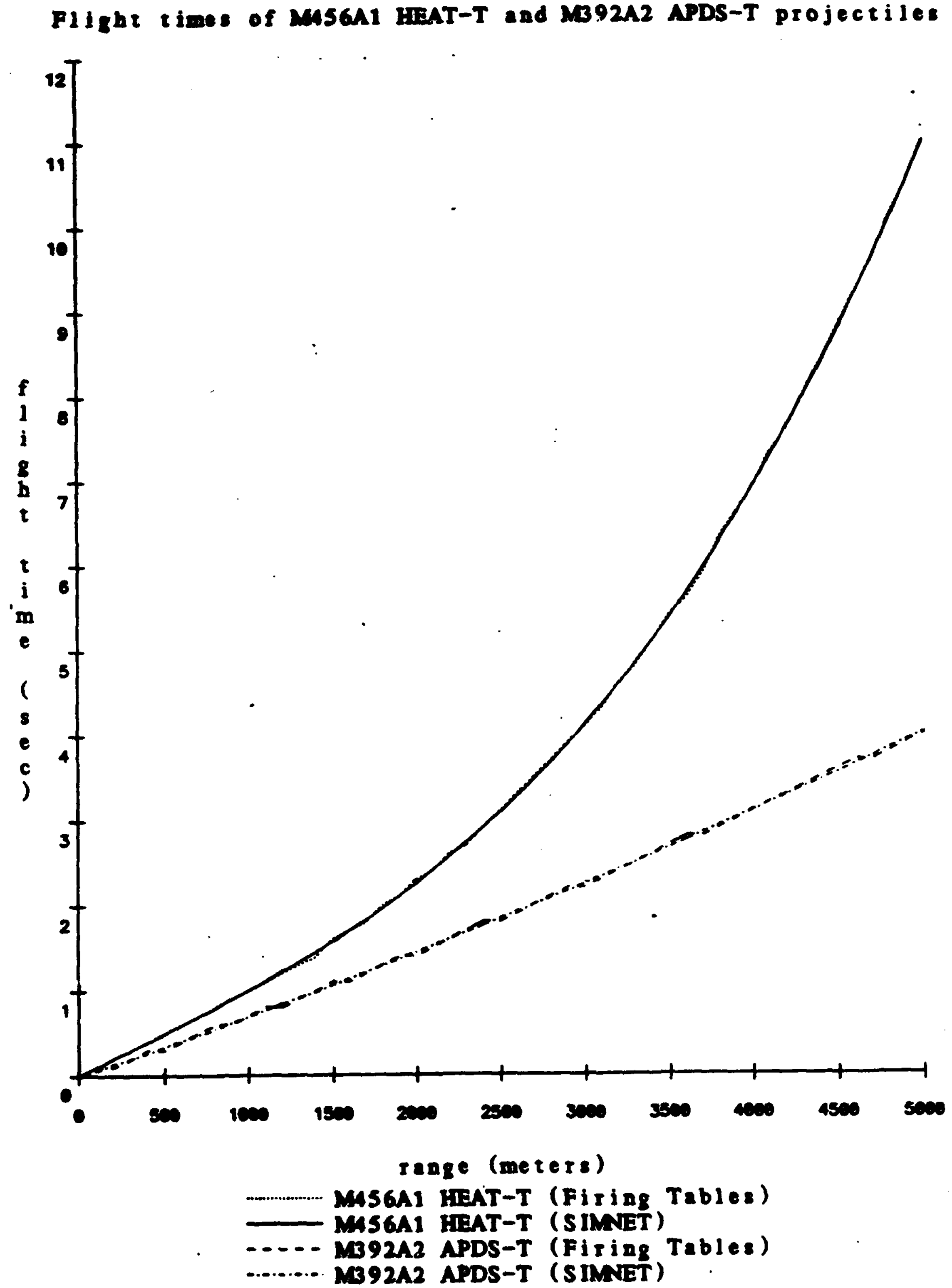
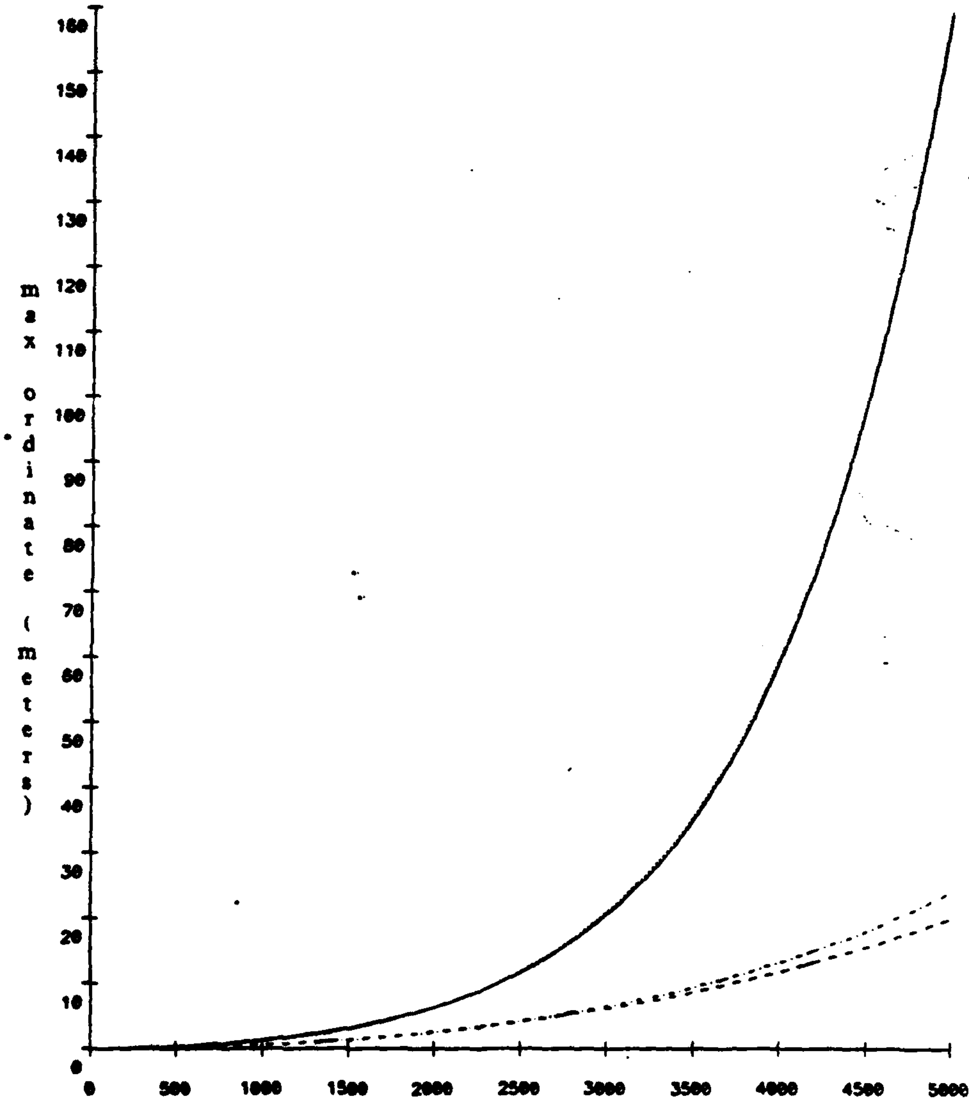


Figure 5-4: Ballistic Validation - Flight time

Maximum ordinates of M456A1 HEAT-T and M392A2 APDS-T trajectories



M456A1 HEAT-T (Firing Tables)
M456A1 HEAT-T (SIMNET)
M392A2 APDS-T (Firing Tables)
M392A2 APDS-T (SIMNET)

Figure 5-5: Ballistic Validation - Maximum Ordinate

12 FT 105-A-3

CARTRIDGE, APDS-T, M392A2 MUZZLE VELOCITY, 1470.20 M/S

RANGE	SUPER-ELEVATION	DX/DSI	DH/DSI	DH/DX	DRIFT	TIME OF FLIGHT
METERS	MILS	M/MIL	M/MIL	M/100M	MILS	SEC
0	0.0	432	0.0	0.0	0.0	0.0
100	0.2	428	0.1	0.0	0.0	0.1
200	0.5	424	0.1	0.0	0.0	0.1
300	0.7	419	0.3	0.1	0.0	0.2
400	0.9	418	0.4	0.1	0.0	0.3
500	1.2	413	0.5	0.1	0.0	0.3
600	1.4	409	0.6	0.1	0.0	0.4
700	1.7	406	0.7	0.2	0.0	0.5
800	1.9	402	0.8	0.2	0.0	0.6
900	2.1	398	0.9	0.2	0.0	0.6
1000	2.4	395	1.0	0.2	0.0	0.7
1100	2.6	392	1.1	0.3	0.1	0.8
1200	2.9	388	1.2	0.3	0.1	0.8
1300	3.1	385	1.3	0.3	0.1	0.9
1400	3.4	382	1.4	0.4	0.1	1.0
1500	3.7	378	1.5	0.4	0.1	1.1
1600	3.9	375	1.6	0.4	0.1	1.1
1700	4.2	372	1.7	0.4	0.1	1.2
1800	4.4	368	1.8	0.5	0.1	1.3
1900	4.7	365	1.9	0.5	0.1	1.4
2000	5.0	362	2.0	0.5	0.1	1.4
2100	5.3	359	2.1	0.6	0.1	1.5
2200	5.5	355	2.2	0.6	0.1	1.6
2300	5.8	352	2.3	0.6	0.1	1.7
2400	6.1	348	2.4	0.7	0.1	1.8
2500	6.4	345	2.5	0.7	0.1	1.8
2600	6.7	342	2.6	0.7	0.1	1.9
2700	7.0	339	2.7	0.8	0.1	2.0
2800	[illegible]	336	2.7	0.8	0.1	2.1
2900	[illegible]	332	2.8	0.9	0.1	2.2
3000	7.8	329	2.9	0.9	0.2	2.2

FT 105-A-3 13

MUZZLE VELOCITY, 1470.20 M/S CARTRIDGE, APDS-T, M392A2

10 KM/HR CROSS WIND DEFLECTION	MAXIMUM ORDINATE	RANGE TO MAXIMUM ORDINATE	ANGLE OF FALL	REMAINING VELOCITY	RANGE
MILS	METERS	METERS	MILS	METERS/SEC	METERS
0.0	0.0	0	0	1470	0
0.0	0.0	51	0	1469	100
0.0	0.0	101	0	1460	200
0.0	0.1	151	1	1451	300
0.0	0.1	202	1	1442	400
0.0	0.1	252	1	1432	500
0.0	0.2	302	1	1423	600
0.0	0.3	353	2	1414	700
0.0	0.4	404	2	1405	800
0.1	0.5	455	2	1396	900
0.1	0.6	506	2	1387	1000
0.1	0.7	557	3	1378	1100
0.1	0.9	608	3	1369	1200
0.1	1.0	660	3	1360	1300
0.1	1.2	711	4	1351	1400
0.1	1.4	763	4	1342	1500
0.1	1.6	814	4	1333	1600
0.1	1.8	866	4	1324	1700
0.1	2.0	918	5	1315	1800
0.1	2.3	970	5	1306	1900
0.1	2.6	1022	5	1297	2000
0.1	2.8	1075	6	1288	2100
0.1	3.1	1127	6	1279	2200
0.2	3.4	1180	6	1270	2300
0.2	3.8	1232	7	1261	2400
0.2	4.1	1285	7	1252	2500
0.2	4.5	1338	7	1243	2600
0.2	4.9	1391	8	1234	2700
0.2	5.3	1444	8	1225	2800
0.2	5.7	1498	8	1216	2900
0.2	6.2	1551	9	1207	3000

14 FT 105-A-3

CARTRIDGE, APDS-T, M392A2 MUZZLE VELOCITY, 1470.20 M/S

RANGE	SUPER-ELEVATION	DX/DSI	DH/DSI	DH/DX	DRIFT	TIME OF FLIGHT
METERS	MILS	M/MIL	M/MIL	M/100M	MILS	SEC
3000	7.8	329	2.9	0.9	0.2	2.2
3100	8.1	326	3.0	0.9	0.2	2.3
3200	8.4	323	3.1	1.0	0.2	2.4
3300	8.7	320	3.2	1.0	0.2	2.5
3400	9.1	317	3.3	1.0	0.2	2.6
3500	9.4	313	3.4	1.1	0.2	2.7
3600	9.7	310	3.5	1.1	0.2	2.8
3700	10.0	307	3.6	1.2	0.2	2.8
3800	10.3	304	3.7	1.2	0.2	2.9
3900	10.7	301	3.8	1.3	0.2	3.0
4000	11.0	298	3.9	1.3	0.2	3.1
4100	11.3	295	4.0	1.4	0.2	3.2
4200	11.6	292	4.1	1.4	0.2	3.3
4300	12.0	289	4.2	1.4	0.2	3.4
4400	12.3	286	4.3	1.5	0.2	3.5
4500	12.7	283	4.4	1.5	0.3	3.6
4600	13.0	280	4.5	1.6	0.3	3.7
4700	13.4	277	4.7	1.7	0.3	3.7
4800	13.7	274	4.7	1.7	0.3	3.8
4900	14.1	271	4.8	1.8	0.3	3.9
5000	14.5	268	4.9	1.8	0.3	4.0

FT 105-A-3 15

MUZZLE VELOCITY, 1470.20 M/S CARTRIDGE, APDS-T, M392A2

10 KM/HR CROSS WIND DEFLECTION	MAXIMUM ORDINATE	RANGE TO MAXIMUM ORDINATE	ANGLE OF FALL	REMAINING VELOCITY	RANGE
MILS	METERS	METERS	MILS	METERS/SEC	METERS
0.2	6.2	1551	9	1207	3000
0.2	6.6	1605	9	1198	3100
0.2	7.1	1658	10	1189	3200
0.2	7.6	1712	10	1181	3300
0.2	8.1	1766	11	1172	3400
0.2	8.7	1820	11	1163	3500
0.2	9.3	1875	11	1154	3600
0.3	9.9	1929	12	1145	3700
0.3	10.5	1984	12	1136	3800
0.3	11.1	2039	13	1128	3900
0.3	11.8	2093	13	1119	4000
0.3	12.5	2148	14	1110	4100
0.3	13.2	2203	14	1101	4200
0.3	13.9	2258	15	1093	4300
0.3	14.7	2314	15	1084	4400
0.3	15.5	2370	16	1075	4500
0.3	16.3	2426	16	1067	4600
0.3	17.2	2481	17	1058	4700
0.4	18.1	2537	17	1049	4800
0.4	19.0	2594	18	1041	4900
0.4	19.9	2650	18	1032	5000

THE FOLLOWING DATA ARE USEFUL WHEN ESTABLISHING RANGE AND MAXIMUM ORDINATE LIMITS FOR CARTRIDGE, APDS-T, M392A2.

ELEVATION - DEGREES	5	10	15	30	57
RANGE - METERS	14950	19570	22885	31064	45722
MAXIMUM ORDINATE - METERS	606	1471	2724	7065	23093

Figure 5-6: Firing Table: M392A2 APDS-T (ref FT-105-A-3)

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
0	0.0	[illegible]	[illegible]	0.0	[illegible]	0.0
100	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
200	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
300	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
400	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
500	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
600	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
700	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
800	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
900	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1000	[illegible]	100	[illegible]	[illegible]	[illegible]	[illegible]
1100	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1200	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1300	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1400	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
1500	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Figure 5-7: Firing Table: M456A1 HEAT-T (ref FT-105-A-3)

6. Electrical System

6.1 Top Level Model Assumptions

The model of the M1 simulation's electrical system is based on a number of assumptions obtained from technical manuals, Army personnel, and engineering judgment. The main assumption is that the charge and voltage are linearly related, and that the battery charges linearly over time. Another assumption is that while many small components (such as indicator lamps) consume electricity, only the major charge consumers (starter, laser range finder, auxiliary hydraulic pump, radio, and intercom) are modeled. Additional knowledge of the actual system includes the following:

- maximum battery voltage is 24 volts,
- maximum battery charge is 300 amp-hours (FULLY_CHARGED state),
- with the engine running, the battery takes 15 minutes to charge up 1 volt,
- alternator provides up to 650 amps, 28 volts at 2400 rpm,
- alternator voltage range is 27.5 to 28.5 volts,
- low battery charge light comes on when battery voltage is < 23 volts (MEDIUM_CHARGED state),
- engine will not start if battery voltage is < 18 volts (WEAKLY_CHARGED state).

6.2 Modes of Operation

The electrical system has several different operating states. They are:

- engine off, master power off
- engine off, master power on
- engine running, alternator working
- engine running, alternator broken

6.2.1 Engine Off Mode

If the tank is shutdown by swiching master power off, the electrical system simulation is disabled. Once master power is turned on, components which discharge the battery are divided into two groups: those which consume electrical charge every frame (e.g. the radio and intercom) and those which consume it only when they are actively in use (e.g. the starter, the laser rangefinder, and the auxiliary hydraulic pump).

For the latter group of components, at the time of the request, the model determines whether or not there is enough charge left in the battery. If so, it sets up a timer to linearly discharge the battery at established rates for the duration of the component's use.

The assumptions made for the starter were:

- The starter could function 40 times without recharging before the engine would no longer start: At this point, the battery would have reached the WEAKLY_CHARGED state and the abort light would become lit on the 41st starting of the engine.
- One starting operation requires 1 second, during which time the battery is linearly discharged.

The characteristics for the 1 hp auxiliary hydraulic pump were conveniently determined by equation. Since the change in charge dQ is defined by the equation $I = \frac{dQ}{dt}$,

$$dQ = Idt, \text{ where}$$

$$I = \frac{746 \text{ watts}}{24 \text{ volts}} = 31.08 \text{ amps, and}$$

$$dt = \frac{1 \text{ sec}}{15 \text{ frames}} \cdot \frac{1 \text{ hour}}{3600 \text{ sec}} = \frac{1 \text{ hr}}{54000 \text{ frames}}$$

Substituting,

$$dQ = 31.08 \text{ amp} \cdot \frac{1 \text{ hr}}{54000 \text{ frames}} = 5.756x10^{-4} \frac{\text{amp-hr}}{\text{frame}}$$

The amount indicated by dQ is discharged each frame the hydraulic pump is in use. The actual duration of use is determined by the hydraulic system.

The last of the components which discharge the battery in discrete quantities is the laser rangefinder. The laser can be fired at most 100 times before the battery is discharged to the WEAKLY_CHARGED state. At this point, neither the starter, the hydraulic pump, nor the laser rangefinder can function. The battery is discharged over one frame time, the one in which the request for usage is made

The radio and intercom are modeled as being on as long as master power is on. Therefore, they discharge the battery a small amount every frame time. They are modeled such that if the radio were the only component on, it could operate for 20 hours before the battery discharged to the WEAKLY_CHARGED state while the intercom could function for 15 hours on its own before reaching that state. While the lamps on the control panels are not modeled individually, if the battery discharges until it has less than 12 volts, any lit lamp is turned off and the electrical system becomes inactive until a maintenance vehicle arrives to recharge (or replace) the battery unit.

6.2.2 Engine Running

The mode in which the engine is running is the simplest to model. If the alternator is functional, it provides enough electrical power to handle all laser rangefinder, hydraulic pump, radio and intercom battery consumption requests, regardless of quantity. It also handles recharging the battery. The current simulation of the alternator recharges the battery at a rate of one volt every three minutes.

The other mode in which the engine is running occurs when the alternator fails. In this state, the engine consumes battery charge such that the battery drops linearly to the WEAKLY_CHARGED state within 5 minutes. Once the battery reaches the WEAKLY_CHARGED state, the engine will spool down, and the driver will be unable to start it again until the alternator and battery are repaired via the Management, Command, and Control (MCC) maintenance console. When the tank does not have an operational alternator, the battery charge consumption requests are handled just as they would be if the engine was off, as described in the preceding section.

The voltmeter displays the alternator voltage if both the engine and alternator are running and functional, and displays the battery voltage in all other cases. The alternator voltage fluctuates between 27.5 and 28.5 volts. It is determined by the engine speed, the limits being 900 rpm at low idle and 3100 rpm at maximum speed. The assumption is that 900 to 3100 rpm corresponds piecewise linearly to the alternator voltage ranges. When driving, the engine is usually operating between 900 and 2000 rpm, so a larger voltage range, 27.5 to 28.2 volts, is devoted to it while the remaining 2000 to 3100 rpm range corresponds linearly to the 28.2 to 28.5 voltage range.

One additional problem which the electrical system can encounter is that of a leaking battery. If the battery leaks, it will charge up to 24 volts (300 amp-hr) with no problems, but it will not hold a charge once a load is placed across it. Thus, if the alternator is not running, the battery will immediately drop to the WEAKLY_CHARGED state. The only corrective action that can be taken for a leaking battery is replacement of the unit.

7. Hydraulic System

7.1 Top Level Model of System

The simulation of the hydraulic system is modeled using data obtained from tests using an M1 tank. The system is pressurized via the main hydraulic pump which is powered directly off the turbine engine, and the electric auxiliary hydraulic pump which operates off battery power for use when the engine is off. The hydraulic pumps draw hydraulic fluid from the 18 gallon reservoir and supply it under pressure to the hull distribution manifold which in turn feeds power distribution valve assemblies in the hull and the turret. These distribution valve assemblies, which regulate the supply pressure to the hydraulic motors and actuators in the hull and turret, are aided by a series of hydraulic accumulators which supplement normal hydraulic pressure during short periods of excessive demand. These accumulators also compensate for fluctuations in hydraulic pressure to provide smooth, even power to the turret drive hydraulic servo motors during target tracking operations.

7.1.1 Hydraulic Pumps

The main hydraulic pump is a variable displacement pump which is driven directly by the accessory drive unit of the turbine engine. With the engine running at normal operational speeds (above 1300 rpm), the main pump will produce between 1550 to 1700 psi at a flow rate of up to 47.4 gal/min (gpm). With the engine at idle (900 rpm), the pump will still deliver 27.1 gpm, which is more than sufficient to meet all the hydraulic needs of the vehicle. The pump regulates its output pressure by varying its internal swash plate angle to adjust for pressure variations due to hydraulic demand on the system.

The auxiliary hydraulic pump is a one horsepower electrically powered unit that runs off the battery to supply hydraulic power when the engine is shut down. This pump is much smaller than the main pump, capable of delivering only 5 gpm at its operating pressure of between 1150 and 1650 psi. The pump is controlled by a pressure sensor in the reservoir which turns the pump on if the pressure in the system drops below 1150 psi, and off once the pressure reaches 1650 psi. The low flow rate of the auxiliary hydraulic pump is not sufficient to meet all the hydraulic needs of the vehicle during a period of heavy hydraulic demand. During a high speed turret traverse, for instance, the turret traverse servo motor will deplete hydraulic pressure much faster than the auxiliary hydraulic pump can rebuild it. If the pressure in the system drops below its minimum working pressure of 800 psi, the

hydraulic servo motor will begin to "cough" and function intermittently due to insufficient supply pressure. The motor will stop working altogether if the pressure drops below 500 psi. In this condition, a crew will either have to wait without hydraulic power until the auxilary hydraulic pump is able to rebuild the reservoir pressure back to a working level, or start the engine to power the main hydraulic pump.

7.2 Lower Level Model Description

The dynamics of the hydraulic system simulation is modeled in an manner similar to that of a charging capacitor,

$$I = C\frac{dV}{dt}$$

where I = current
C = capacitance
V = voltage.

The analogous hydraulic relationship is:

$$Q = K\frac{dP}{dt}$$

where Q = volume flow rate
K = hydraulic capacitance
P = pressure.

The hydraulic capacitance K of the hydraulic system was empirically derived from the observation that it took 45 seconds for the auxiliary hydraulic pump (with a flow rate of 5 gpm) to raise the system pressure from 500 psi to 1500 psi. Hence,

$$K \cdot \frac{1000\ \text{psi}}{45\ \text{sec}} = \frac{5\ \text{gal}}{\text{min}}$$

$$K = \frac{0.225\ \text{gal-sec}}{\text{psi-min}} = \frac{0.00375\ \text{gal}}{\text{psi}}$$

The next parameter that needed to be defined was the flow rate Q_T of the turret traverse hydraulic servo motor at its maximum speed. This was derived from the observation that during a full speed traverse with the engine shut

off. it took 6.0 seconds for the pressure to drop from 1500 psi to 900 psi and that the motor began to function intermittently. Hence,

$$Q_T = K \cdot \frac{(1500-900)\ \text{psi}}{6.0\ \text{sec}}$$

Substituting the value of K derived earlier,

$$Q_T = 22.5 \frac{\text{gal}}{\text{min}}$$

Hence, the turret traverse hydraulic servo motor flow of hydraulic fluid is 22.5 $\frac{\text{gal}}{\text{min}}$ when the turret is slewing at its maximum rate.

7.3 Components Modeled

The hydraulic operations chosen to be modeled in this simulation were the slewing of the turret, elevation of the gun, opening and closing of the ammo door and setting the parking brake. The maximum flow rate Q_T = 22.5 gpm of the traverse hydraulic servo motor was also used to describe the maximum flow rate Q_E of the hydraulic actuator used to elevate the main gun. For both actuators, the percentage of the 22.5 gpm max flow rate which applies is based on the turret slew rate represented as a percentage of its maximum slewing rate, or likewise, the gun elevation rate as a percentage of the gun's maximum rate. The hydraulic system is called every frame that the turret slews or the gun elevates using hydraulic pressure.

The ammunition door and parking brake are modeled by simply decrementing the hydraulic reservoir by discrete quantities of hydraulic pressure each time the ammo door opens or closes or the parking brake is set. In addition, the ammo door uses a small amount of pressure for each frame time the door remains open. If the pressure drops below 750 psi, the ammo door will not open and likewise, if the pressure drops below 800 psi, the parking brake can not be set.

8. Depletable Resource Management

8.1 Fuel

Fuel management is divided into three subsystems: fuel tanks, fuel transfer and fuel resupply.

8.1.1 Fuel Tanks

The fuel system simulation models the three fuel tanks in the M1: the left front tank, the right front tank, and the rear tank. The rear tank has a fuel capacity of 248.0 gal while the left and right front tanks can hold 106.6 gal and 149.8 gal respectively for a total fuel capacity of 504.4 gal. At any time, the fuel level meter will display the quantity of fuel in the tank indicated by the fuel tank selector switch.

Although the M1 has 3 fuel tanks, the engine only draws fuel from the rear tank. The amount of fuel used each frame (1/15 sec) is determined by the engine model. The fuel low light illuminates when the rear tank's fuel level drops to 1/4 full. Once the fuel low light is on, it will *not* turn off again until the rear tank is at least 3/8 full. As the rear fuel tank becomes depleted, the driver must transfer fuel from either of the front tanks to the rear tank. If all three tanks are emptied, the engine simulation will not function, effectively immobilizing the vehicle until it is refueled from an M559 fuel service truck dispatched from the MCC Admin/Log Center.

8.1.2 Fuel Transfer

Fuel transfer can occur between the front and rear tanks but not between the left and right front tanks. Two conditions must be met before fuel can be transferred -- master power must have been on for more than five seconds (to insure that accidental fuel transfer cannot take place immediately after master power is turned on), and the fuel low light must be on. Once these conditions are satisfied, fuel transfer will begin as soon as the fuel tank selector switch is set to a front tank. Fuel will be transferred at a rate of 30 gpm. When the tank becomes 3/4 full, fuel transfer will cease. If during fuel transfer, the tank selector switch is turned back to *rear tank* after the fuel low light has turned off (rear tank is at least 3/8 full), fuel cannot be transferred again until the fuel low light becomes relit. In actuality, a real tank crew should transfer fuel from the right front tank first because the personnel heater uses fuel from the left front tank. Since our simulation of the M1 does not model the personnel heater, this concern does not

apply. If a fuel transfer pump failure occurs, no additional fuel can be transferred, but the engine can still draw fuel from the remaining fuel in the rear tank.

8.1.3 Fuel Resupply

Since fuel is a depletable resource, it is necessary to simulate its resupply from a fuel carrier. To resupply fuel, the resupply officer in the Admin/Log Center dispatches an M559 fuel carrier to a specified location. If the M1 tank drives close enough to the M559 and satisfies a few other conditions, fuel will be transferred at a rate of 30 gallons per minute from the M559 to the tank.

When the commander wishes to transfer fuel, he instructs his driver to approach the fuel carrier. If the tank is alive and stationary, the fuel supply is not full, and there is a fuel carrier within 20 meters, the tank requests fuel from the fuel carrier by sending a message over the network to the MCC host. The MCC responds by sending back a message indicating that the fuel carrier is ready to transfer fuel. At that point, the tank will begin to take on fuel at a rate of 30 gallons per minute until the fuel resupply conditions are not satisfied. The fuel resupply software attempts to transfer fuel into the rear tank, then into the right front tank if the rear tank is full, and finally into the left front tank if the other tanks are full. While the fuel transfer is taking place, the LOW FUEL lamp will flash once per second, and if the tank selector switch is turned to the tank which is being refueled, the fuel meter will slowly rise. Every 40 seconds, the data structures are updated to note the transferral taking place.

See figure 8-1 for an illustration of the fuel resupply finite state machine. The three fuel resupply states are QUIET, REQUEST, and LOADING, and the finite state machine operates as follows:

QUIET STATE:
If the following conditions are ALL true:
1. The tank is alive.
2. The tank is not moving.
3. There are M559 fuel carriers within a distance of 20 meters.
4. The fuel load in the tank is not full.

Then
1. The simulator sends a service request packet to the MCC.
2. The simulator starts a 5 second timer.
3. The simulator enters the REQUEST STATE.

REQUEST STATE:
If AT LEAST ONE of the following conditions is true:
1. The tank is dead.
2. The tank is moving.
3. There are no M559 fuel carriers within a distance of 20 meters.
4. The fuel load in the tank is full.

Then

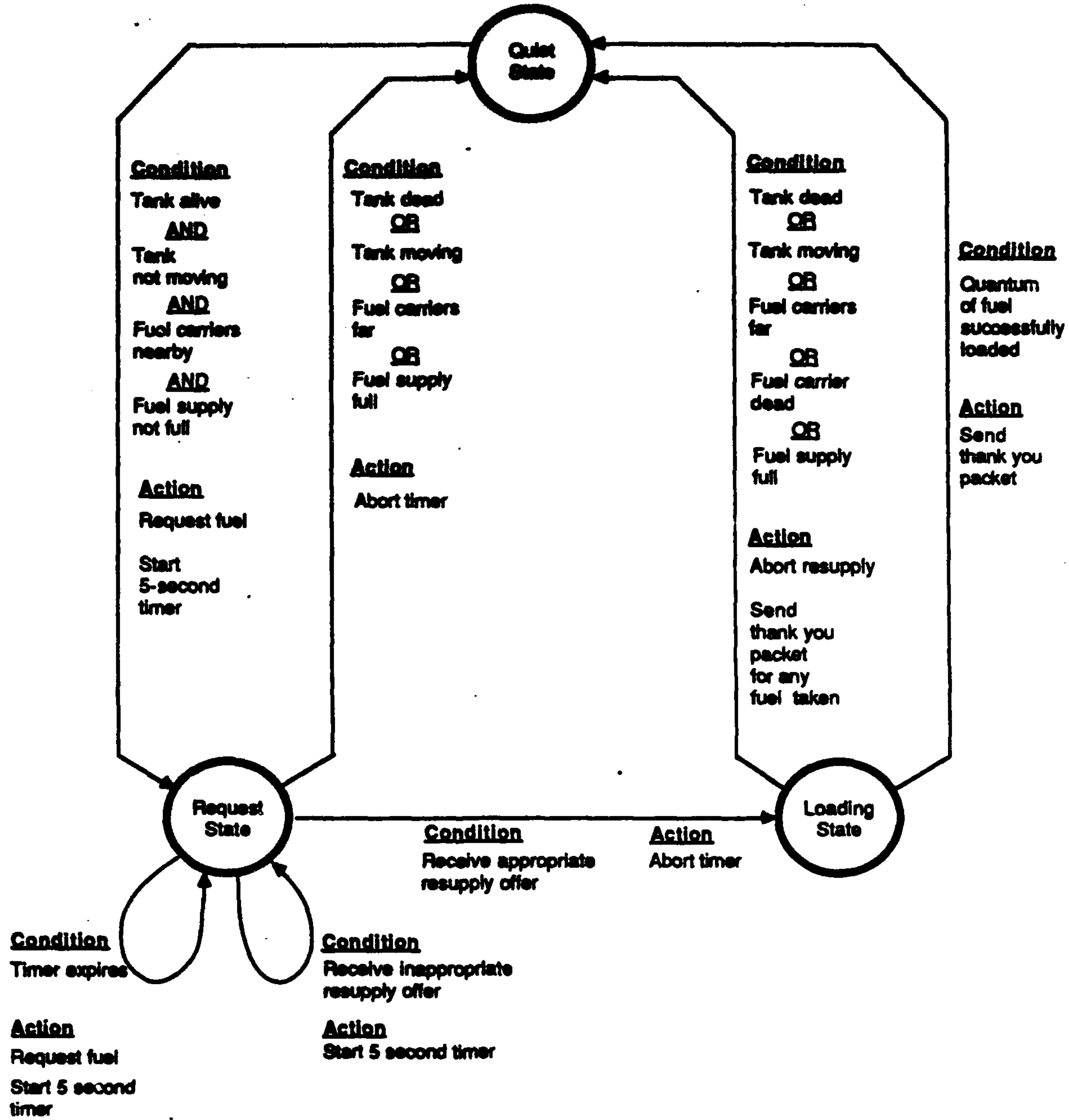


Figure 8-1: Fuel Resupply Finite State Machine

1. The 5 second timer is aborted.
2. The simulator enters the QUIET STATE.

If the 5 second timer expires:
Then

1. The simulator sends another service request packet to the MCC.
2. The simulator starts a 5 second timer.
3. The simulator stays in the REQUEST STATE.

If the simulator receives an appropriate resupply offer from the MCC:
Then
1. The timer is aborted.
2. The simulator enters the LOADING STATE.

If the simulator receives an inappropriate resupply offer from the MCC:
Then:
1. The simulator starts a 5 second timer.
2. The simulator stays in the REQUEST STATE.

LOADING STATE:
If AT LEAST ONE of the following conditions is true:
1. The tank is dead.
2. The tank is moving.
3. There are no M559 fuel carriers within a distance of 20 meters.
4. The fuel carrier which offered fuel is dead.
5. The fuel load in the tank is full.

Then
1. The resupply is aborted.
2. The simulator sends a thank-you packet to the MCC which records the amount of fuel taken before the resupply was aborted.
3. The simulator enters the QUIET STATE.

If 20 gallons of fuel have been successfully loaded in 40 seconds:
Then
1. The simulator sends a thank-you packet to the MCC which records the amount of fuel taken.
2. The simulator enters the QUIET STATE.

By means of this finite state machine protocol and a similar one on the MCC. both the simulator and the MCC will maintain an accurate account of the their respective fuel supplies.

8.2 Ammunition

The ammunition subsystem of the M1 simulation is responsible for the storage, transfer, and resupply of ammunition. The relevant physical controls for the ammunition subsystem are the READY rack which consists of 22 ammunition buttons each of which is equipped with a SABOT lamp and a HEAT lamp, the knee switch, the ejection guard, the breech button and breech unload switch, and the ammunition select switch and lamps. The M1 tank has a capacity of 55 rounds of ammunition: 22 stowed in the READY rack, 22 stowed in the SEMI-READY rack, and a total of 11 rounds stowed in HULL/TURRET FLOOR (representing the 8 rounds in the rear of the crew compartment, and the 3 rounds stowed on the turret basket floor to the left of the main gun).

8.2.1 Initialization

The initial ammunition supply is provided during the MCC vehicle activation sequence. The activation packet received by the simulation specifies the initial number of HEAT-T and APDS-T rounds in each of the three ammunition locations, and these numbers are recorded for bookkeeping and resupply purposes. The rounds allocated to the READY rack are assigned slots and the corresponding READY rack data structure is updated.

8.2.2 Loader's Sequence

The ammunition software is designed to enforce the proper sequence of actions which the loader must perform to successfully transfer ammunition around the turret. If he wishes to load a round from the READY rack to the breech, he first presses the knee switch, which uses the hydraulic system to open the thick steel blast door which protects the rounds in the READY rack. When the blast door opens, the contents of the 22 READY rack slots are revealed: each slot which contains a APDS-T round will turn on its SABOT lamp, and each slot which contains a HEAT-T round will turn on its HEAT lamp. If he presses the button on a slot which contains a round, the lamp on that slot turns off. This simulates the movement of the round into the loader's arms. He may put the round back into the READY rack by pressing the button on an empty slot. Two seconds after the loader releases the knee switch, the blast door closes and all the lamps turn off. The loader must set the ejection guard switch to the SAFE position in order to load the round into the breech. When the loader's arms are full, the breech is empty, the ejection guard is SAFE, and 2.5 seconds have elapsed since the loader released the knee switch, the "READY" lamp underneath the breech button turns on, which means that the breech is ready to be loaded. When the loader presses the breech

button, the round is loaded into the breech, a sound is produced, the "READY" lamp turns off, and the "LOADED" lamp turns on. If the loader's arms are empty, the breech is full, the ejection guard is SAFE, and the loader is not transferring or resupplying ammunition, he may unload the round from the breech into his arms by pulling the unload breech switch.

If the loader fails to follow the proper loading/unloading sequence, the ammunition software responds appropriately. If he attempts to remove a round from an empty slot or to insert a round into a full slot, the ammunition subsystem will not change its state. Also, if he tries to load a round into the breech when one of the ready conditions is not satisfied, the round will stay in his arms.

8.2.3 Ammunition Transfer

Only 22 of the 55 rounds of ammunition in the M1 tank are located in the READY rack. There are a maximum of 22 rounds in the SEMI-READY rack behind the commander, and a maximum of 11 rounds on the hull and turret floors. Since the loader does not have easy access to the rounds in the SEMI-READY rack and on the floors, it is necessary to transfer these rounds to the READY rack before using them. This procedure takes 40 seconds per round, and usually takes place when the tank is concealed or in a strategically safe position.

When the commander wishes to transfer a round from within the tank, he turns his ammunition transfer switch to either SEMI-READY HEAT, SEMI-READY SABOT, HULL HEAT or HULL SABOT depending on the type of round and the location from which he wants to transfer. When the ammunition switch is moved into place, the blast door is automatically opened and the contents of the READY rack are revealed. To transfer a round, the loader presses the button on an empty slot. While the 40-second transfer is taking place the HEAT or SABOT lamp (depending on the round chosen) on that slot and also the appropriate lamp above the ammunition transfer switch will flash once per second to display the source and destination of the round. If the ammunition transfer switch is moved during this time, the transfer is aborted. If the loader presses the button on another empty slot during this time, then the HEAT or SABOT lamp on that slot will continue the flashing. After 40 seconds, the lamp is turned on and the data structures are updated to note the transferral. When the commander returns the ammunition transfer switch to NO TRANSFER, the blast door will close after two seconds.

If the loader or commander fails to follow the proper transfer sequence, the ammunition software responds appropriately. If one attempts to transfer a round into a full slot, the round is not transferred. If the ammunition

transfer switch is turned to HULL HEAT and there are no heat rounds on the hull/turret floor, then the HULL HEAT lamp will turn off and no round will be transferred. In addition, the transferral will not take place if the loaders arms are already full.

8.2.4 Ammunition Resupply

Since ammunition is a depletable resource, it is necessary to simulate its resupply from an ammunition carrier. The MCC may dispatch ammo carriers to specified locations and if the M1 tank drives close enough to one and satisfies a few other conditions, a round will be transferred in 40 seconds from the ammo carrier to the M1 tank.

When the commander wishes to transfer a round from an ammo carrier, he turns his ammunition transfer switch to REDIST RECV (to receive a round from another vehicle). When the ammunition switch is moved into place, the blast door is automatically opened and the contents of the READY rack are revealed. If the tank is alive and stationary, the ammo load is not full, the loader's arms are empty, and there is an ammo carrier within 20 meters, the tank requests ammunition from the ammo carrier by sending a network message to the MCC host, which responds with a message indicating that the ammo carrier is ready to transfer ammunition. The tank resupplies one round at a time, 40 seconds per round, until the ammunition resupply conditions are not satisfied. The ammunition resupply software attempts to place each round in the READY rack, then in the SEMI-READY rack if the READY rack is full, and finally on the hull/turret floor if both READY racks are full. The software also tries to maintain the original ratio of HEAT to SABOT, and chooses the round which will produce a ratio closer to the original. While the 40-second transfer is taking place the REDIST RECV lamp and either the HULL HEAT lamp, HULL SABOT lamp, SEMI-READY HEAT lamp, SEMI-READY SABOT lamp, or the appropriate lamp on a slot in the READY rack will flash once per second to display the source and destination of the round. If the ammunition transfer switch is moved during this time, the transfer is aborted. If the loader presses the button on an empty READY rack slot during this time, then the HEAT or SABOT lamp on that slot will continue the flashing. After 40 seconds, the lamp is turned on and the data structures are updated to note the transferral. When the commander returns the ammunition transfer switch to NO TRANSFER, the blast door will close after two seconds.

See figure 8-2 for an illustration of the ammunition resupply finite state machine. The three ammunition resupply states are QUIET, REQUEST, and LOADING, and the finite state machine operates as follows:

QUIET STATE:

If the following conditions are ALL true:

1. The tank is alive.
2. The tank is not moving.

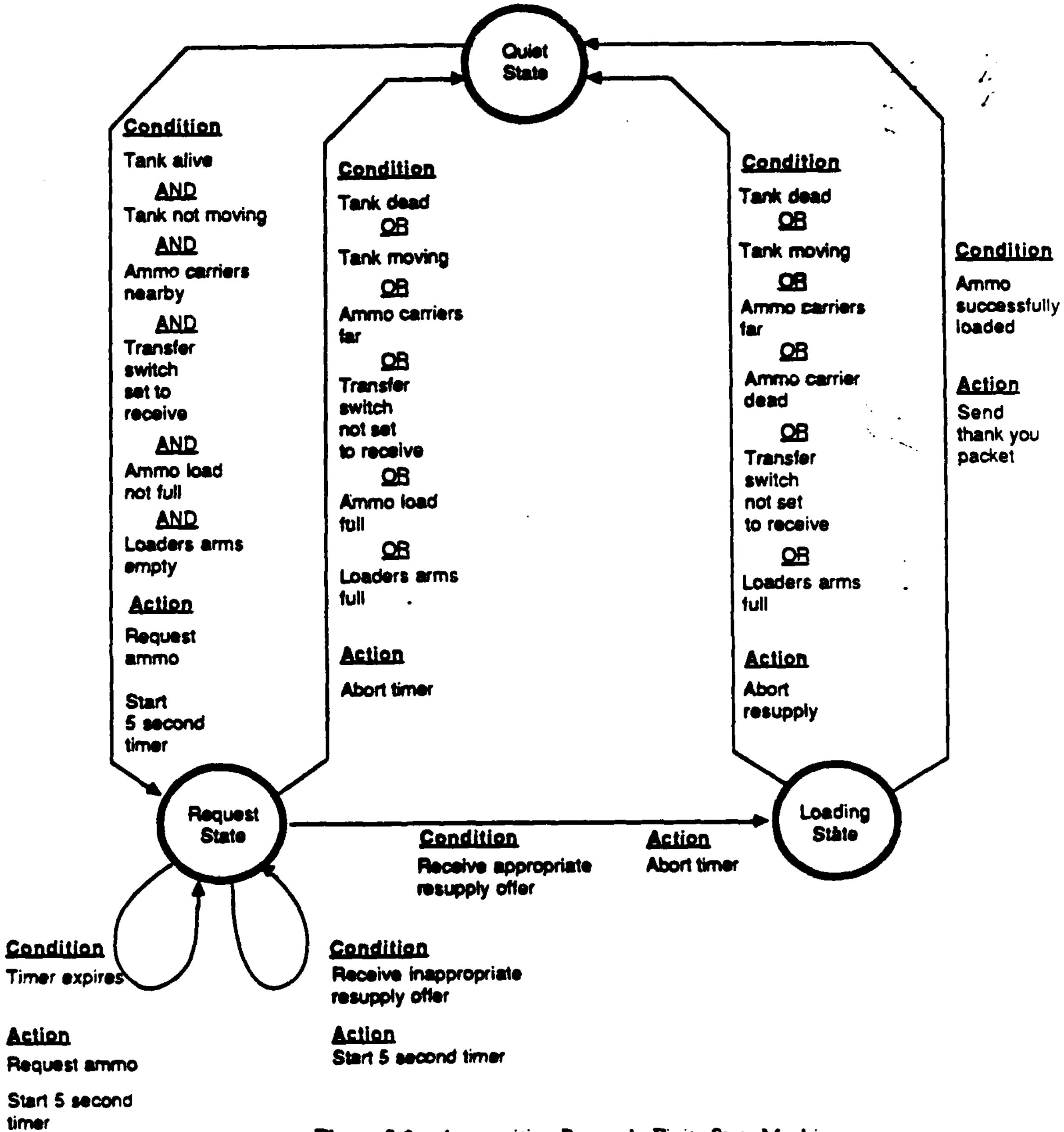


Figure 8-2: Ammunition Resupply Finite State Machine

3. There are ammunition carriers within a distance of 20 meters.
4. The commander has set the ammunition transfer switch to REDIST_RECV.
5. The ammunition load in the tank is not full.
6. The loader's arms are not empty.

Then

1. The simulator sends a service request packet to the MCC.
2. The simulator starts a 5 second timer.
3. The simulator enters the REQUEST STATE.

REQUEST STATE:

If AT LEAST ONE of the following conditions is true:

1. The tank is dead.
2. The tank is moving.
3. There are no ammunition carriers within a distance of 20 meters.
4. The commander has not set the ammunition transfer to REDIST_RECV.
5. The ammunition load in the tank is full.
6. The loader's arms are full.

Then

1. The 5 second timer is aborted.
2. The simulator enters the QUIET STATE.

If the 5 second timer expires:

Then

1. The simulator sends another service request packet to the MCC.
2. The simulator starts a 5 second timer.
3. The simulator stays in the REQUEST STATE.

If the simulator receives an appropriate resupply offer from the MCC:

Then

1. The timer is aborted.
2. The simulator enters the LOADING STATE.

If the simulator receives an inappropriate resupply offer from the MCC:

Then:

1. The simulator starts a 5 second timer.
2. The simulator stays in the REQUEST STATE.

LOADING STATE:

If AT LEAST ONE of the following conditions is true:

1. The tank is dead.
2. The tank is moving.
3. There are no ammunition carriers within a distance of 20 meters.
4. The ammunition carrier which offered ammunition is dead.
5. The commander has not set the ammunition transfer switch to REDIST_RECV.
6. The loader's arms are full.

Then

1. The resupply is aborted.
2. The simulator enters the QUIET STATE.

If the ammunition is successfully loaded after 40 seconds:

Then

1. The simulator sends a thank-you packet to the MCC records the type of ammunition taken.
2. The simulator enters the QUIET STATE.

By means of this finite state machine protocol and a similar one on the MCC, both the simulator and the MCC

host maintain consistent records of the exact inventory of rounds. The protocol is designed to ensure that all rounds in SIMNET are accounted for, with none being dropped, and no "extras" being created due a communications error.

9. Crew Station Controls and Displays

The particular controls and displays provided in the SIMNET M1 simulators were selected because they were necessary to accomplish the tasks designed for training. The controls and displays omitted from the simulators are pictured in their actual locations to ensure the proper orientation of crew members. The functionality of the simulated controls is designed to reflect that of the actual controls in the M1 tank. There are a few controls and displays in the simulators which are designed to provide information which would be obvious in the actual M1 tank or to simulate crew member actions which would not involve physical controls.

9.1 Panels and IDC Boards

The controls and displays in the M1 simulators are divided into five stations corresponding to the commander, driver, gunner, loader, and ammunition rack. Each station consists of a few panels with buttons, switches, lamps, meters, and/or physical devices (such as the driver's service brake or the loader's periscope) which resemble those in an actual M1 tank. See Sections 9.4-9.8 for listings of the controls and displays simulated by SIMNET.

Each station is serviced by a custom-made Interactive Device Controller (IDC) board which is drives all the controls and displays for that station. The IDC board communicates with the simulation host via an RS232 connection, and therefore acts as an intermediary between the physical controls and the data used by the simulation.

Each state transition of a physical control causes the memory in the IDC board to be updated. The Z80 processor on the IDC board loops forever through its memory checking for physical controls which have just changed their state, and when it discovers one it sends a 4-byte message to the simulation host on the RS232 line (2 bytes of header, followed by 1 byte each for control identifier and new value). Conversely, if the simulation wishes to toggle a lamp, or to change the state of a meter, the simulation host sends a 4-byte message to the IDC board on the RS232 line, which updates the IDC board memory and causes the output device to be changed.

9.2 Simulation Peripherals and Shared Memory

The messages from the IDC board to the simulation host are received by the High Performance Serial Multiplexer (HPSM), a card which resides on the simulation host Multibus. The HPSM and the M1 simulation process run independently, and communicate with each other via shared memory. The HPSM process automatically starts whenever the simulation host is rebooted, and runs asynchrously from the simulation process.

The 68010 processor in the HPSM card loops 14 times per second through its internal memory checking for control identifiers which have just changed their value, and when it discovers one it writes the new value into a large memory array shared by the HPSM and the simulation processes. The controls software reads this memory array 15 times per second and uses these values to determine the state of the physical controls on the M1 simulator. Conversely, if the simulation wishes to send a message to the IDC board, it enqueues a message onto a large memory queue shared by the simulation and the HPSM processes. The HPSM process reads this queue 14 times per second, creates 4-byte messages, and writes them to the RS232 line connected to the IDC board. This method of queueing is advantageous because the real-time simulation process does not need to wait on UNIX system calls.

9.3 Controls Software

The controls software in the M1 simulation reads the shared memory corresponding to all the IDC boards 15 times per second, and uses this information to direct the simulation of the various M1 subsystems. Conversely, the controls software is called by the M1 subsystems to produce outputs and responds by writing output messages onto the shared memory queues.

The controls software is organized into a finite state machine model, where the three states are based on the power status of the M1 simulator: no power, only master power, and turret power (see figure 9-1).

Transitions between states are directed by changes in the Master Power Switch and the Turret Power Switch and electrical failures and repairs. Several actions occur during state transitions. For example, when turret power is turned on, several status lights are restored or turned on, stabilization system gyroscopes are spooled up, the ballistics computer is reset, and several turret subsystems are initialized. The reasoning behind the finite state machine model is that the various controls in the tank may be classified into three categories: those that operate

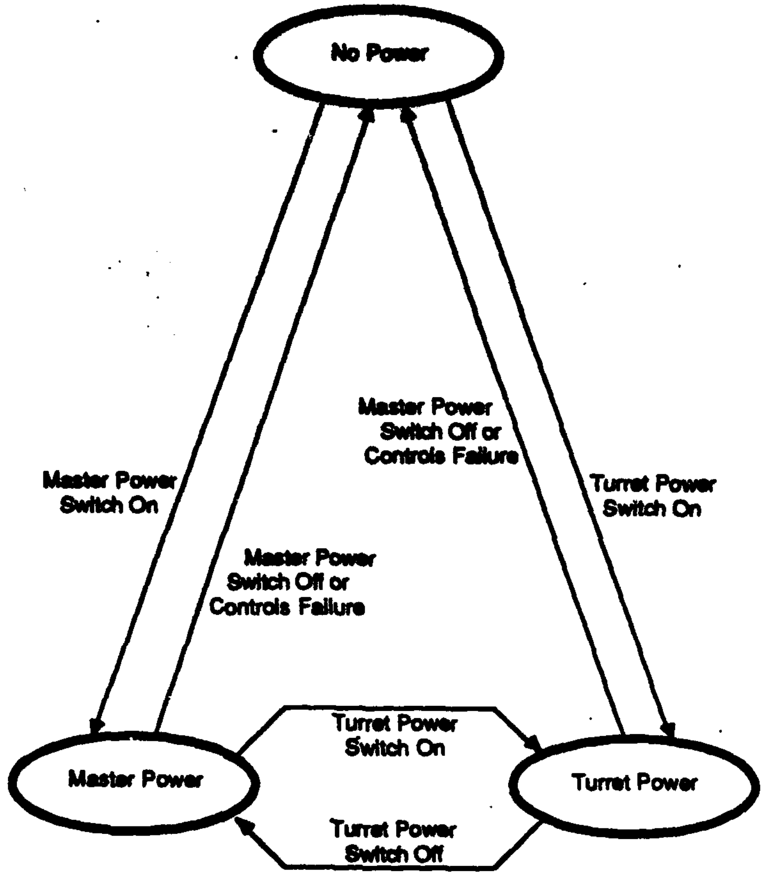


Figure 9-1: Controls Finite State Machine

regardless of the power state, those that need master power to operate, and those that need turret power to operate. The controls software only reads from the controls and panels which can affect the simulation based on the current power state. See Section 9.9 for listings of the controls and panels sampled by the controls software in the three states of the finite state machine.

When the finite state machine is in the NO POWER state, the MASTER POWER and TURRET POWER controls are effectively cut off from supplying input to the simulation, and likewise when in the MASTER POWER state, the TURRET POWER controls are ignored. Although this separation of M1 simulator controls into different power classifications does not speed up the worst case situation which occurs when turret power is on, it organizes the code in a very efficient, highly reliable manner.

During each 1/15 second interval, the controls software samples each control in turn and calls procedures in the various M1 simulated subsystems if the control has changed its state. The subsystems may in turn call the controls software if an output display must be updated.

The controls inputs are classified into binary level inputs, potentiometer level inputs, and edge inputs. The binary level inputs include the switches and buttons of which the current value is needed by the simulation. The multiple-value switches, such as the transmission and the ammunition transfer switch, consist of multiple binary level inputs where the input corresponding to a particular value is set HIGH when the multiple-value switch would have that value. The potentiometer level inputs include the steering bar, service brake, and the turret traversal and elevation handles. The potentiometer values written into shared memory are scaled into real numbers between either 0.0 and 1.0 or -1.0 and 1.0 before these values are passed to the various subsystems. The edge inputs include the breech button and the engine start switch, i.e., those inputs of which only the LOW-HIGH transition is needed by the simulation.

The controls outputs are classified into binary outputs and meter outputs. The binary outputs are the lamps in the various crew stations, while the meter outputs are all located in the driver's station. The output displays are highly dependent on the power state organization, since the hull lamps should not be turned on when master power is off, and the turret lamps should not be turned on when turret power is off. Therefore, for each lamp, there is a shared memory location which records which state the lamp would have if power were on. When a subsystem wishes to turn on a lamp, the controls software writes a LAMP_ON code into the shared memory location but actually turns on the lamp only if power is on.

The controls software is subject to simulated electrical failure and catastrophic kill failure. When the electrical system fails, the controls finite state machine causes a transition to the NO POWER state just as though turret power and master power were turned off but also disables the master and turret power switches. When the electrical system is repaired, the master and turret power switches are enabled once again. When the controls suffer a catastrophic kill failure, the finite state machine resets all the IDC boards, turns off every lamp, jumps to the NO POWER state, and ignores all the controls.

9.4 Commander Station IDC Controls and Displays

Turret Traverse Handle
Gun Elevate Handle
Cupola
Master Power Switch
Turret Power Switch
Manual Range Battlesight Button
Manual Range Add/Drop Switch
Panel Test Button
Laser Button
Trigger
Palm Switch
Ammunition Transfer Switch
Grid Azimuth Push Button
Master Power Lamp
Turret Power Lamp
Low Battery Lamp
Fire Control Malfunction Lamp
Ammunition Transfer Indicator Lamps

9.5 Driver Station IDC Controls and Displays

Steering Bar
Throttle
Service Brake
Master Power Switch
Panel Test Button
Engine Start Button
Tactical Idle Switch
Engine Shutoff Switch
Parking Brake
Parking Brake Release Switch
Transmission
Warning/Caution Lamp Reset Switch
Fuel Tank Selector Switch
Master Power Lamp
Engine Abort Lamp
Engine Start Lamp
Brake Lamp
Master Warning Lamp
Master Caution Lamp
Engine Oil Temperature High Lamp
Engine Oil Pressure Low Lamp
Transmission Oil Temperature High Lamp
Transmission Oil Pressure Low Lamp
Low Battery Lamp
Low Fuel Lamp
Engine Overspeed Lamp

Engine Oil Low Lamp
Transmission Oil Low Lamp
Engine Filter Clogged Lamp
Transmission Filter Clogged Lamp
Fuel Filter Clogged Lamp
Right Fuel Pump Inoperative Lamp
Left Fuel Pump Inoperative Lamp
Meter Lamps
Tachometer
Speedometer
Voltmeter
Fuel Gauge

9.6 Gunner Station IDC Controls and Displays

Turret Traverse Handle
Gun Elevate Handle
Fire Control Switch
Gun Select Switch
Ammunition Select Switch
GPS Magnification Switch
Laser Select Switch
Laser Button
Trigger
Palm Switch
Fire Control Normal Lamp
Fire Control Emergency Lamp
Gun Select Main Lamp
Gun Select Trigger Safe Lamp
Ammo Select Heat Lamp
Ammo Select Sabot Lamp
Turret To Hull Indicator Lamps

9.7 Loader Station IDC Controls and Displays

Periscope
Breech Button
Breech Unload Switch
Ejection Guard Switch
Gun/Turret Drive Switch
Main Gun Armed Lamp
Main Gun Safe Lamp
Breech Ready Lamp
Breech Loaded Lamp
Power Drive Lamp
Elevation Uncoupled Lamp
Manual Drive Lamp

9.8 Ammunition Rack Station IDC Controls and Displays

22 Round Select Buttons
Knee Switch
22 Heat Ready Lamps
22 Sabot Ready Lamps

9.9 FSM State Definitions

The following tables detail which input functions in the simulator are enabled and active in each state of the finite-state machine:

9.9.1 NO POWER STATE

Parking Brake
Parking Brake Release Switch
Service Brake
GPS Magnification Switch
Ejection Guard Switch
Breech Button
Breech Unload Switch
Ammunition Transfer Switch
Knee Switch
22 Round Select Buttons
Commander's Cupola
Loader's Periscope
Grid Azimuth Push Button

9.9.2 MASTER POWER ON STATE

All No Power Controls
Steering Bar
Throttle
Transmission
Fuel Tank Selector Switch
Engine Start Button
Engine Shutoff Switch
Tactical Idle Switch
Warning/Caution Lamp Reset Switch
Driver's Panel Test Button

9.9.3 TURRET POWER ON STATE

All No Power Controls
All Master Power Controls
Fire Control Switch
Gun/Turret Drive Switch
Gun Select Switch
Gunner's Palm Switch
Commander's Palm Switch
Gunner's Turret Traverse Handle
Commander's Turret Traverse Handle
Gunner's Gun Elevate Handle
Commander's Gun Elevate Handle
Gunner's Laser Button
Commander's Laser Button
Gunner's Trigger
Commander's Trigger
Laser Select Switch
Manual Range Battlesight Button
Manual Range Add/Drop Switch
Ammunition Select Switch
Commander's Panel Test Button

10. Damage and Failure Simulation

The failure module allows the introduction of simulated physical failures and degradations of a tank's capabilities. Part of this section involves the presentation of the failure to the crew and to the other simulation modules. Lights, gauges, sounds, and CIG displays are used to inform the crew of simulated failures. Various routines are called to direct other simulation modules to use either normal or degraded functionality.

The failures are divided into categories that are indicative of the method used for failure generation. The three categories are:

- combat damage
- stochastic failure
- deterministic failure

Combat damage can be inflicted when the tank receives direct fire hit data or indirect fire data from the network. The location of the hit, the type of ammunition used, and various probabilities of damage determine which failures occur.

A stochastic failure occurs when a tank subsystem fails on its own and not because of a crew error or due to combat damage. The frequency of a subsystem failure is determined by its Mean Number of Operations Between Failure (MNOBF). Currently, mileage is used as the unit of Operation. Stochastic failures can degrade functions or can serve as a warning for potential deterministic failures.

Deterministic failures are those failures which result from some improper action that the crew has taken and generally could have been prevented by the crew. These include failures due to resource depletion and failures due to crew error. Examples of these errors include mismanaging fuel and ammunition, ignoring warning lights, throwing a track as a result of driving the vehicle across a hill with too great a slope or turning too quickly at high speed in soft soil.

10.1 Combat Damage

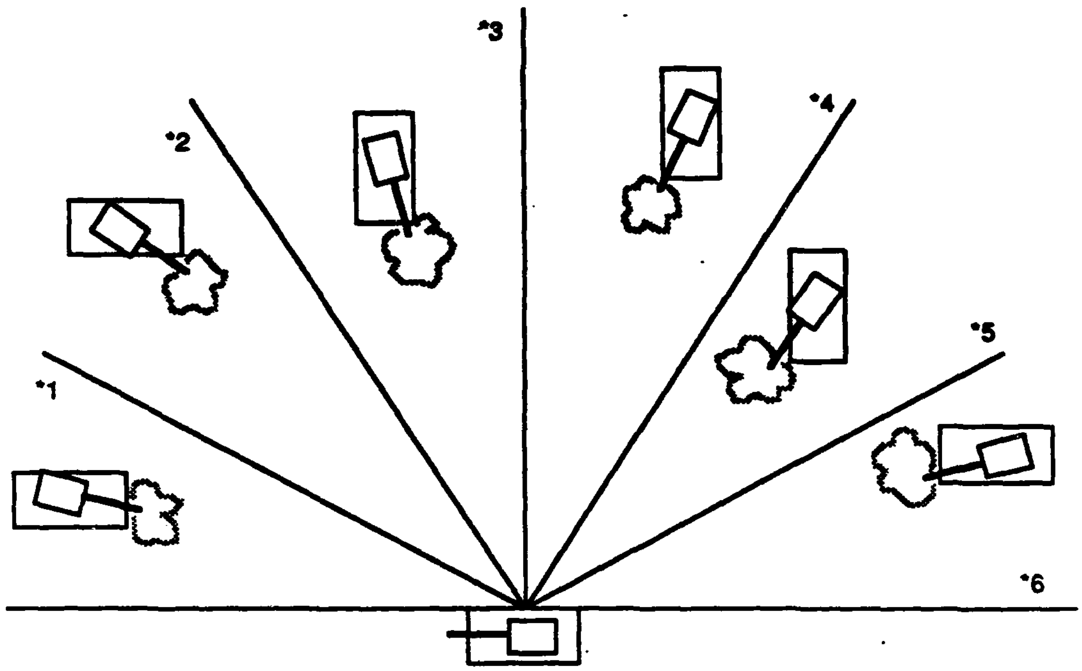
During combat the vehicle receives combat information messages from the network. This information comes in three different forms:

1. A **vehicle hit message** tells the tank that it has been hit by an incoming direct fire round. This message also tells the vehicle what type of round hit it (either HEAT or SABOT), whether the round hit the hull or turret, which side of the vehicle the round hit, and what the angle of incidence of the incoming round was.
2. A **ground impact message** tells the vehicle about a direct fire round that missed any targets, and hit the ground instead. This information is passed to the CIG system to display the impact, and is passed to the sound system to generate an appropriate impact sound if the vehicle is close enough to the impact point.
3. An **indirect fire message** tells the vehicle that an indirect fire round has landed on the terrain. It contains information that tells where the round hit, and what type of ammunition landed. The impact point is used to determine if the impact was close enough to damage the tar f so, the location of the impact point and the tank location and headings are used to determine what part of the tank was exposed to the round. Information from the message is passed to the CIG and sound systems to generate indirect fire explosions and sounds.

When a vehicle determines that it may have been damaged by a round (either a direct fire round or an indir fire round that landed close to the vehicle), then it must determine where on the vehicle the damage may ha occured. Several factors are used to determine where the damage may have occured, including the side of t vehicle hit, the direction that the round came from, the angle of incidence of the incoming round, and whether round hit the hull or turret. For each hit location, there are several possible failures that may occur. If a failure d occur (based on a probability given for each failure in the given location), then a routine is called to notify appropriate module(s) affected by the failure.

10.1.1 Direct Fire

The following tables indicate the direct fire damage probabilities of the M1 as implemented for the SIM simulator. The variable parameters that determine the direct fire damage probabilities are direction of impact or turret impact, ammunition type, and angle of incidence.



HEAT Hits to M1 Hull
Right Side

*1
Right Front 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
80% right track loss

*2
Right Front 30-60° damages
30% kill - blast doors closed
100% kill - blast doors open
30% right track loss
30% engine loss
30% transmission loss

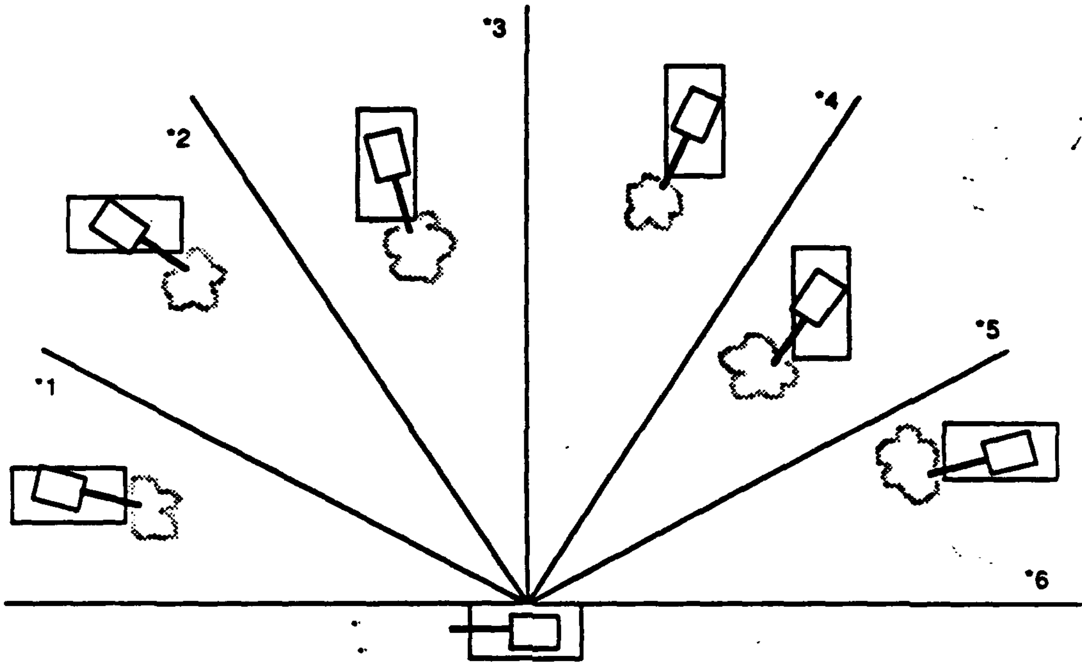
*3
Right Front 00-30° damages
40% kill - blast doors closed
100% kill - blast doors open
10% right track loss

*4
Right Back 00-30° damages
40% kill blast door closed
100% kill blast door open
10% right track loss

*5
Right Back 30-60° damages
30% kill blast door closed
100% kill blast door open
30% right track loss
30% engine loss
30% transmission loss

*6
Right Back 60-90° damages
10% kill blast doors closed
100% kill blast doors open
80% right track loss

Figure 10-1: Direct Fire Damage: HEAT hit - Hull Right Side



SABOT Hits to M1 Hull Right Side

*1
Right Front 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
50% right track loss
20% engine loss
20% transmission loss

*2
Right Front 30-60° damages
30% kill - blast doors closed
100% kill - blast doors open
20% right track loss
20% engine loss
20% transmission loss

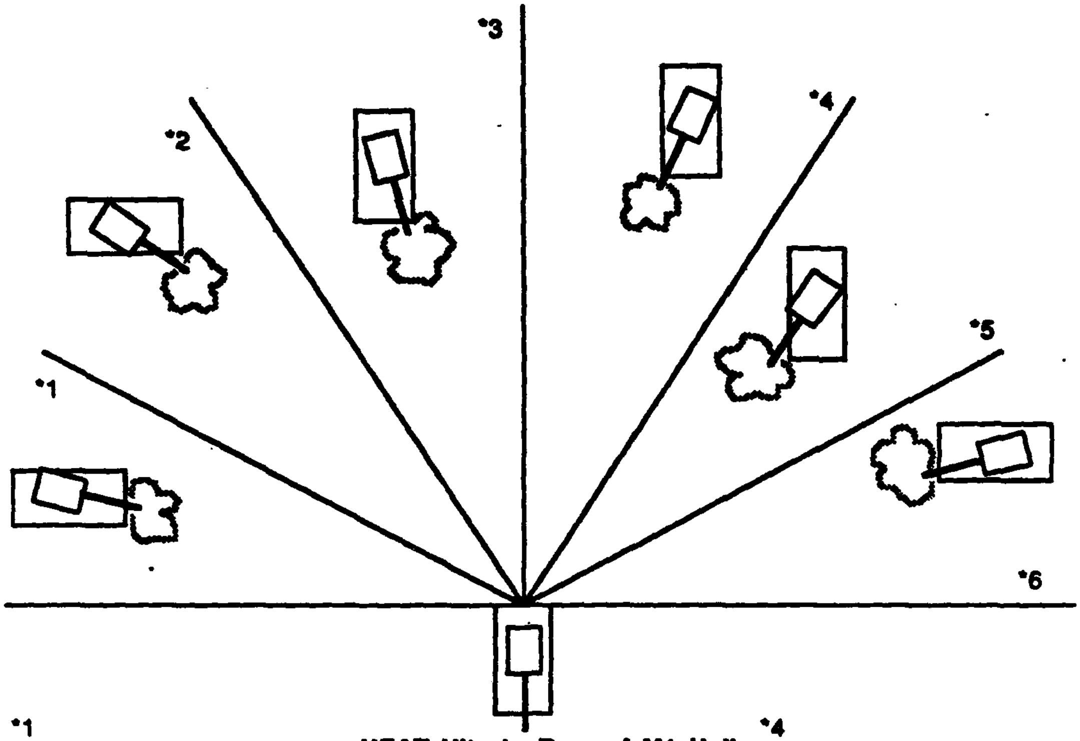
*3
Right Front 00-30° damages
40% kill - blast doors closed
100 % kill - blast doors open
20% engine loss
20% transmission loss

*4
Right Back 00-30° damages
40% kill blast door closed
100% kill blast door open
20% engine loss
20% transmission loss

*5
Right Back 30-60° damages
30% kill blast door closed
100% kill blast door open
20%right track loss
20% engine loss
20% transmission loss

*6
Right Back 60-90° damages
10% kill blast doors closed
100% kill blast doors open
50% right track loss
20% engine loss
20% transmission loss

Figure 10-2: Direct Fire Damage: APDS hit - Hull Right Side



HEAT Hits to Rear of M1 Hull

*1
Back Right 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
90% left track loss
40% engine loss
40% transmission loss

*2
Back Right 30-60° damages
30% kill - blast doors closed
100% kill - blast doors open
50% left track loss
50% engine loss
50% transmission loss

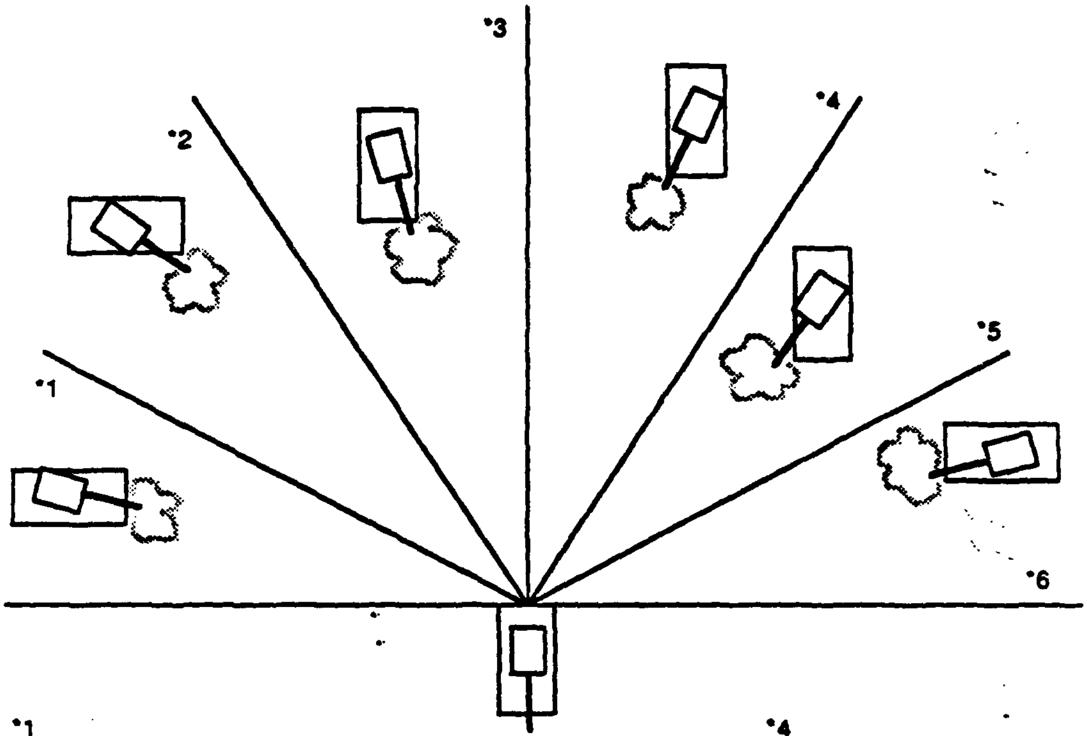
*3
Back Right 00-30° damages
60% kill - blast doors closed
100% kill - blast doors open
30% left track loss
60% engine loss
60% transmission loss

*4
Back Left 00-30° damages
60% kill blast door closed
100% kill blast door open
30% right track loss
60% engine loss
60% transmission loss

*5
Back Left 30-60° damages
30% kill blast door closed
100% kill blast door open
50%right track loss
50% engine loss
50% transmission loss

*6
Back Left 60-90° damages
10% kill blast doors closed
100% kill blast doors open
90% right track loss
40% engine loss
40% transmission loss

Figure 10-3: Direct Fire Damage: HEAT hit - Hull Rear



SABOT Hits to Rear of M1 Hull

*1
Back Right 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
60% left track loss
40% engine loss
40% transmission loss

*2
Back Right 30-60° damages
30% kill - blast doors closed
100% kill - blast doors open
40% left track loss
80% engine loss
80% transmission loss

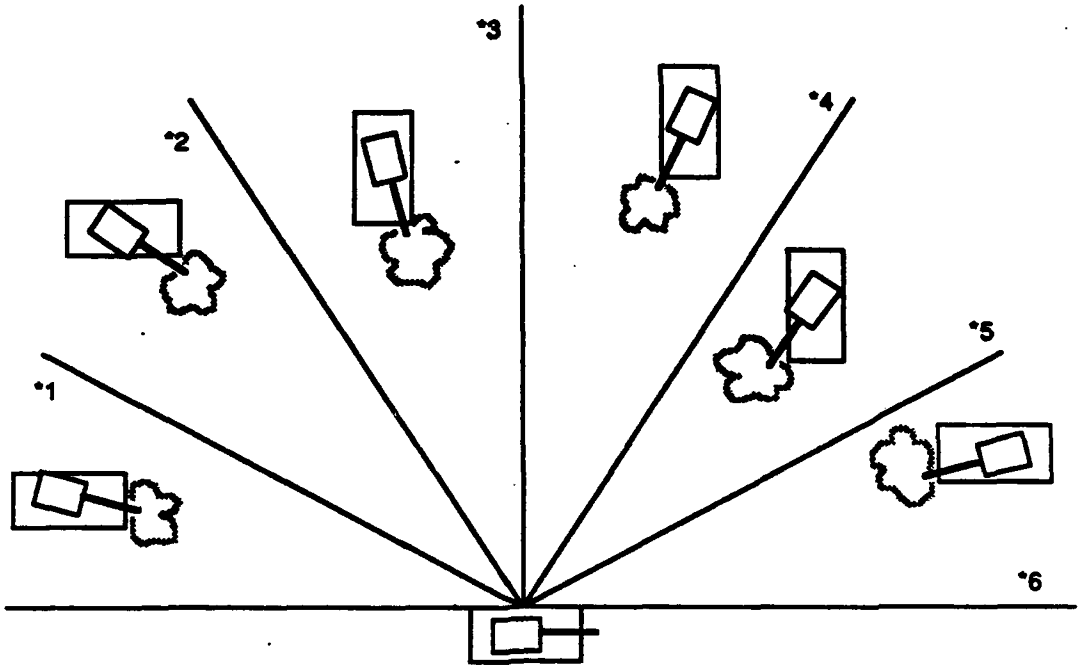
*3
Back Right 00-30° damages
60% kill - blast doors closed
100% kill - blast doors open
30% left track loss
100% engine loss
100% transmission loss

*4
Back Left 00-30° damages
60% kill blast door closed
100% kill blast door open
30% right track loss
100% engine loss
100% transmission loss

*5
Back Left 30-60° damages
30% kill blast door closed
100% kill blast door open
40%right track loss
80% engine loss
80% transmission loss

*6
Back Left 60-90° damages
10% kill blast doors closed
100% kill blast doors open
60% right track loss
40% engine loss
40% transmission loss

Figure 10-4: Direct Fire Damage. APDS hit - Hull Rear



HEAT Hits to M1 Hull
Left Side

*1
Left Back 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
80% left track loss

*2
Left Back 30-60° damages
30% kill - blast doors closed
100% kill - blast doors open
30% left track loss
30% engine loss
30% transmission loss

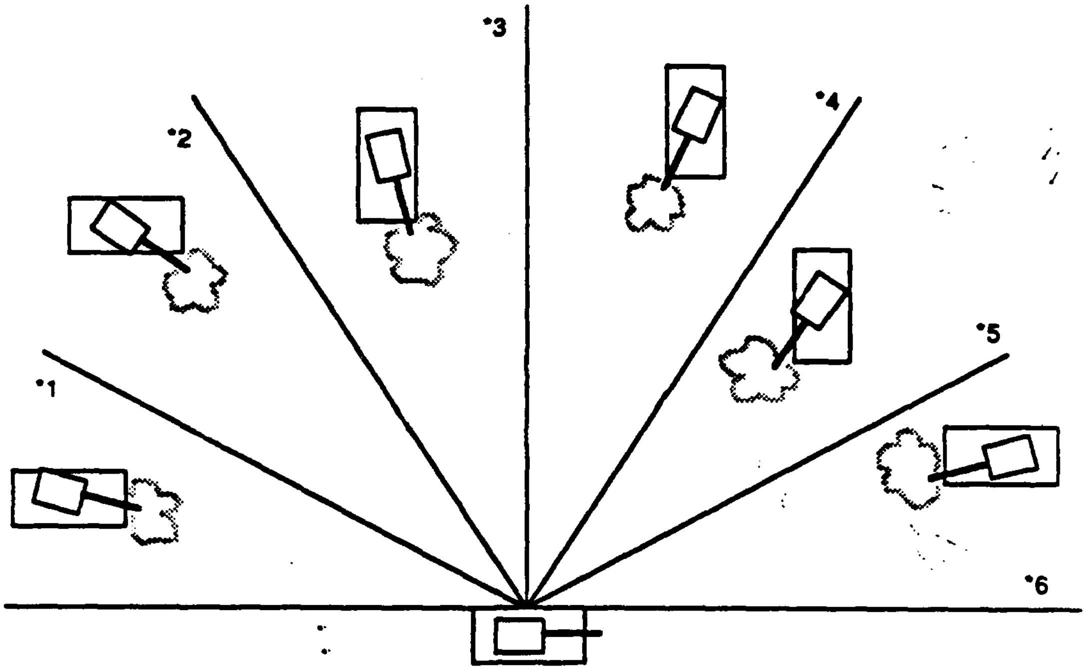
*3
Left Back 00-30° damages
40% kill - blast doors closed
100% kill - blast doors open
10% left track loss

*4
Left Front 00-30° damages
40% kill blast door closed
100% kill blast door open
10% left track loss

*5
Left Front 30-60° damages
30% kill blast door closed
100% kill blast door open
30% left track loss
30% engine loss
30% transmission loss

*6
Left Front 60-90° damages
10% kill blast doors closed
100% kill blast doors open
80% left track loss

Figure 10-5: Direct Fire Damage: HEAT hit - Hull Left Side



SABOT Hits to M1 Hull
Left Side

*1
Left Back 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
50% left track loss
20% engine loss
20% transmission loss

*2
Left Back 30-60° damages
30% kill - blast doors closed
100% kill - blast doors open
20% left track loss
20% engine loss
20% transmission loss

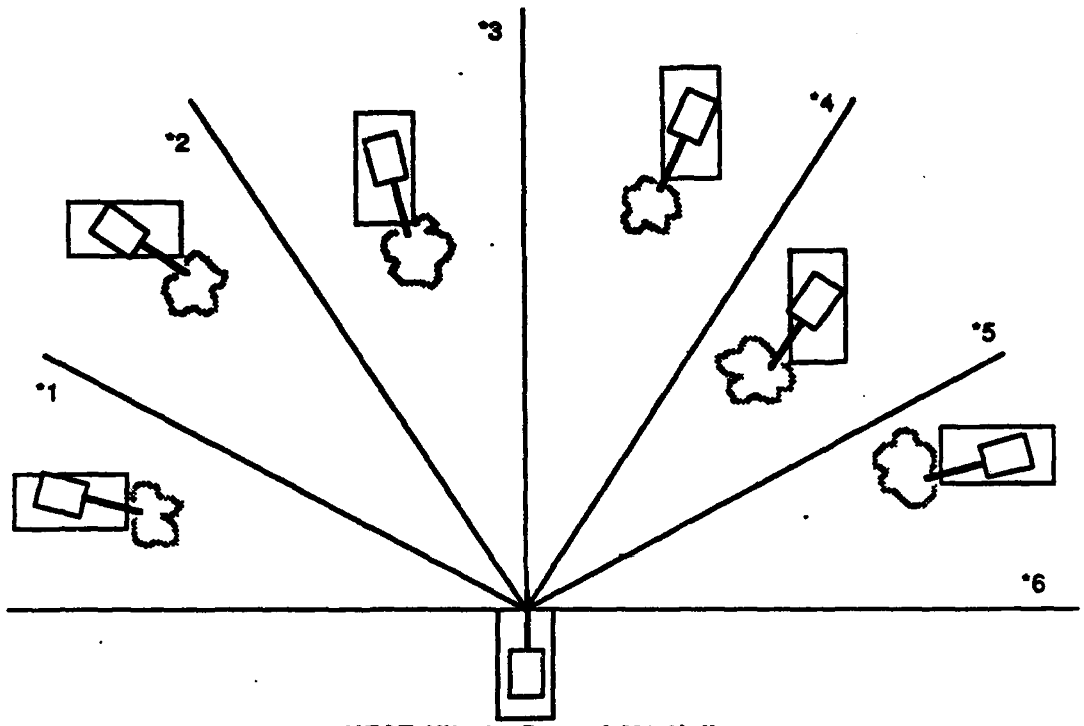
*3
Left Back 00-30° damages
40% kill - blast doors closed
100% kill - blast doors open
20% engine loss
20% transmission loss

*4
Left Front 00-30° damages
40% kill blast door closed
100% kill blast door open
20% engine loss
20% transmission loss

*5
Left Front 30-60° damages
30% kill blast door closed
100% kill blast door open
20% left track loss
20% engine loss
20% transmission loss

*6
Left Front 60-90° damages
10% kill blast doors closed
100% kill blast doors open
50% left track loss
20% engine loss
20% transmission loss

Figure 10-6: Direct Fire Damage: APDS hit - Hull Left Side



*1
Front Left 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
90% right track loss

*2
Front Left 30-60° damages
15% kill - blast doors closed
100% kill - blast doors open
50% right track loss

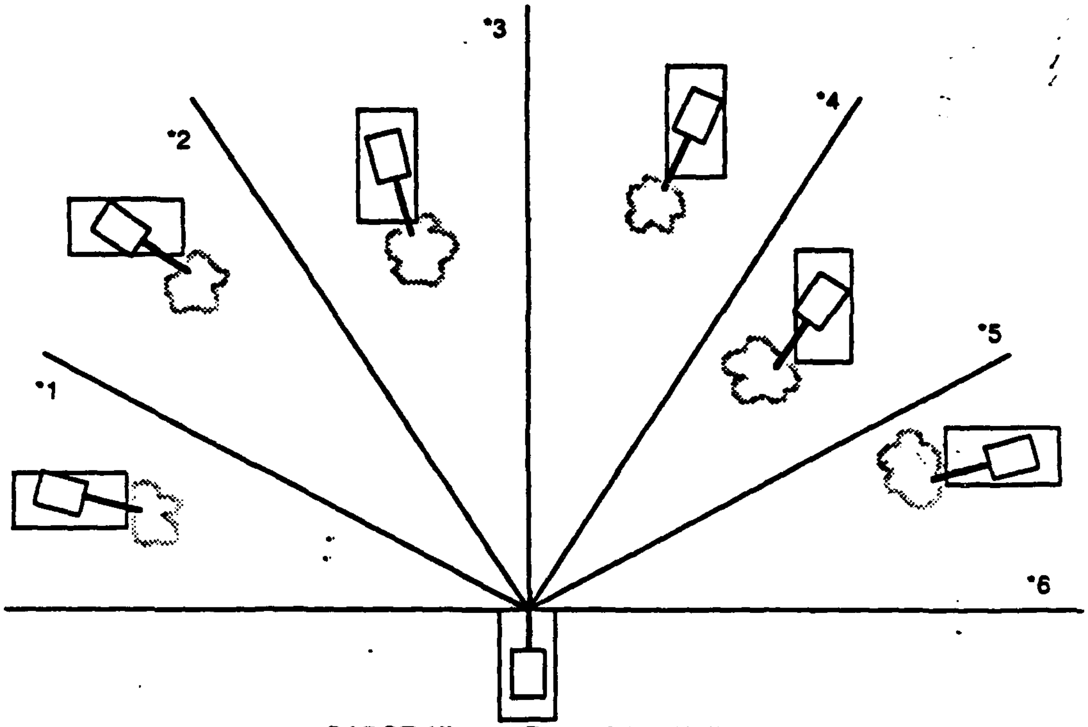
*3
Front Left 00-30° damages
20% kill - blast doors closed
100% kill - blast doors open
30% right track loss
60% driver vision block loss

*4
Front Right 00-30° damages
20% kill blast door closed
100% kill blast door open
30% left track loss
60% driver vision block loss

*5
Front Right 30-60° damages
15% kill blast door closed
100% kill blast door open
50% left track loss

*6
Front Right 60-90° damages
10% kill blast doors closed
100% kill blast doors open
90% left track loss

Figure 10-7: Direct Fire Damage: HEAT hit - Hull Front



SABOT Hits to Bow of M1 Hull

*1
Front Left 60-90° damages
10% kill - blast doors closed
100% kill - blast doors open
60% right track loss

*2
Front Left 30-60° damages
15% kill - blast doors closed
100% kill - blast doors open
40% right track loss

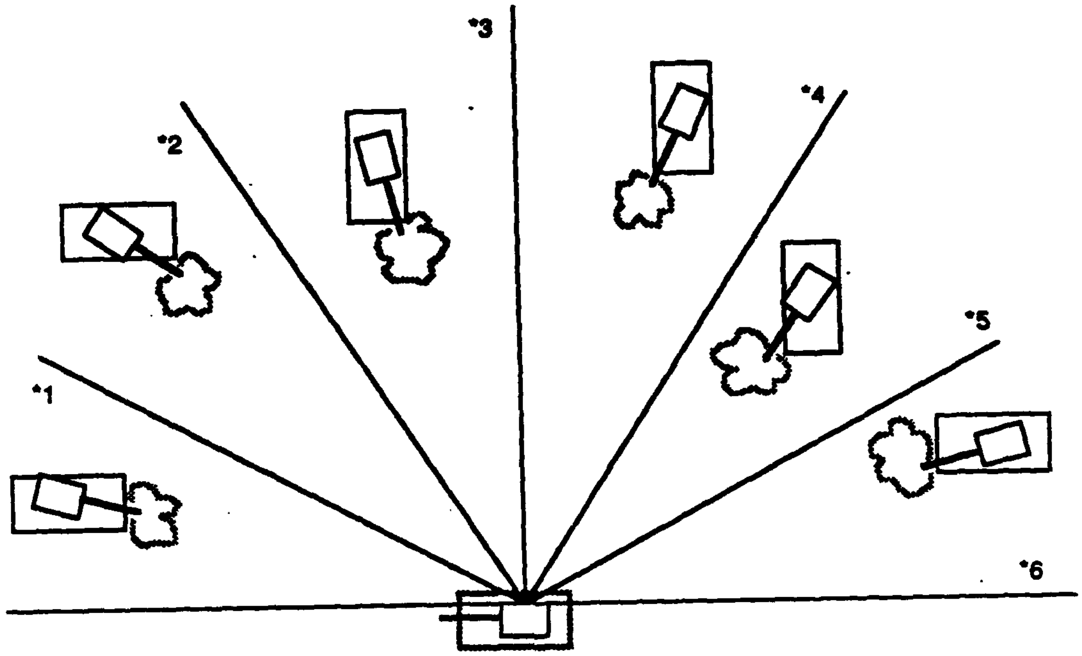
*3
Front Left 00-30° damages
20% kill - blast doors closed
100% kill - blast doors open
30% right track loss
40% driver vision block loss

*4
Front Right 00-30° damages
20% kill blast door closed
100% kill blast door open
30% left track loss
40% driver vision block loss

*5
Front Right 30-60° damages
15% kill blast door closed
100% kill blast door open
40% left track loss

*6
Front Right 60-90° damages
10% kill blast doors closed
100% kill blast doors open
60% left track loss

Figure 10-8: Direct Fire Damage: APDS hit - Hull Front



HEAT Hits to Right Side
of M1 Turret

*1
Right Front 60-90° damages
30% kill - blast doors closed
100% kill - blast doors open
10% turret gearbox

*2
Right Front 30-60° damages
40% kill - blast doors closed
100 % kill - blast doors open
20% gun mount / turret interface
20% turret gearbox

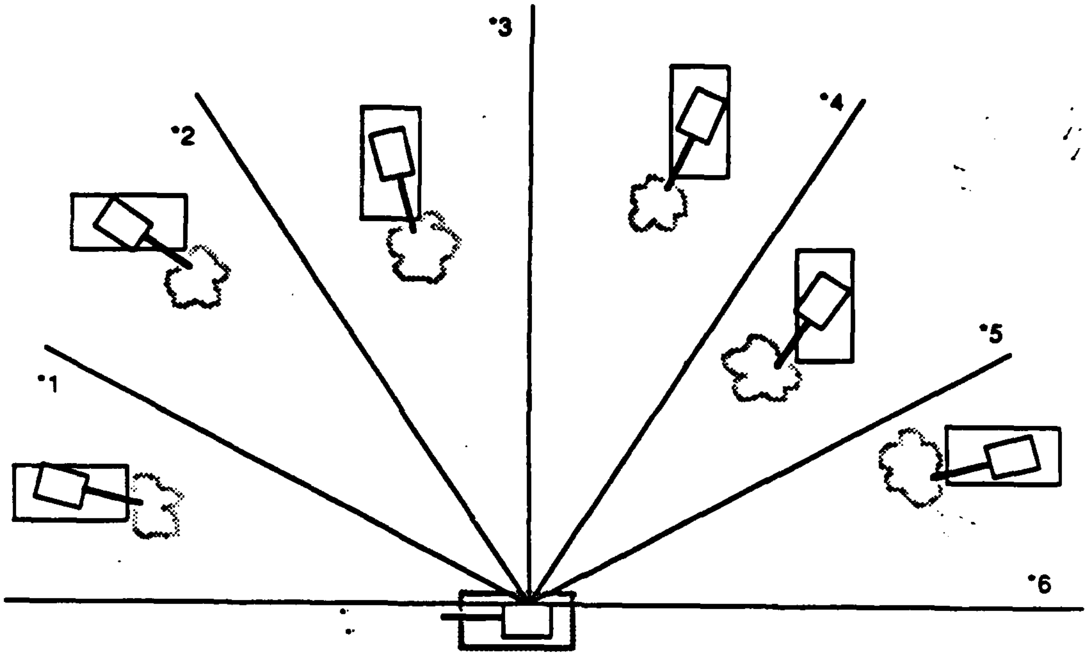
*3
Right Front 00-30° damages
50% kill - blast doors closed
100% kill - blast doors open
20% LRF
20% GPS
30% gun mount / turret interface
50% turret gearbox

*4
Right Back 00-30° damages
50% kill blast door closed
100% kill blast door open
20% LRF
20% GPS
30% gun mount / turret interface
50% turret gearbox

*5
Right Back 30-60° damages
40% kill blast door closed
100% kill blast door open
20% gun mount / turret interface
20% turret gearbox

*6
Right Back 60-90° damages
30% kill blast doors closed
100% kill blast doors open
10% turret gearbox

Figure 10-9: Direct Fire Damage: HEAT hit - Turret Right Side



SABOT Hits to Right Side of M1 Turret

*1
Right Front 60-90° damages
30% kill - blast doors closed
100% kill - blast doors open
20% gun mount / turret interface
10% turret gearbox

*2
Right Front 30-60° damages
40% kill - blast doors closed
100 % kill - blast doors open
20% gun mount / turret interface
10% turret gearbox

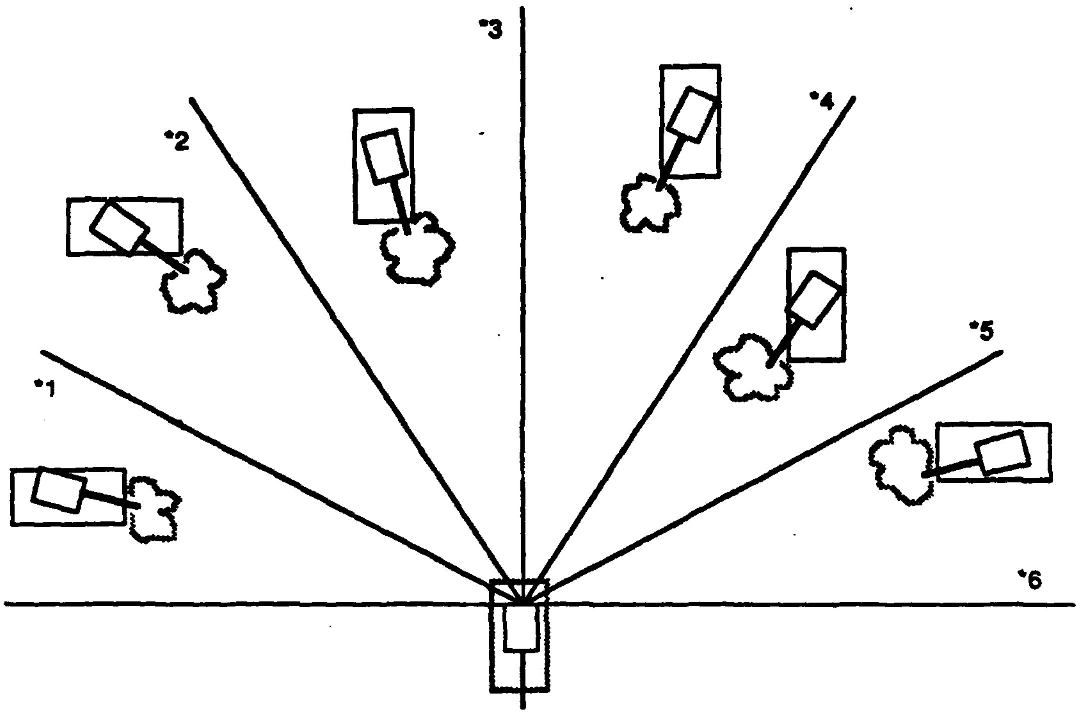
*3
Right Front 00-30° damages
50% kill - blast doors closed
100% kill - blast doors open
20% LRF
20% GPS
30% gun mount / turret interface
40% turret gearbox

*4
Right Back 00-30° damages
50% kill blast door closed
100% kill blast door open
20% LRF
20% GPS
30% gun mount / turret interface
40% turret gearbox

*5
Right Back 30-60° damages
40% kill blast door closed
100% kill blast door open
20% gun mount / turret interface
10% turret gearbox

*6
Right Back 60-90° damages
30% kill blast doors closed
100% kill blast doors open
20% gun mount / turret interface
10% turret gearbox

Figure 10-10: Direct Fire Damage: APDS hit - Turret Right Side



HEAT Hits to Rear of
M1 Turret

*1
Back Right 60-90° damages
40% kill - blast doors closed
100% kill - blast doors open

*2
Back Right 30-60° damages
60% kill - blast doors closed
100 % kill - blast doors open
10% turret gearbox

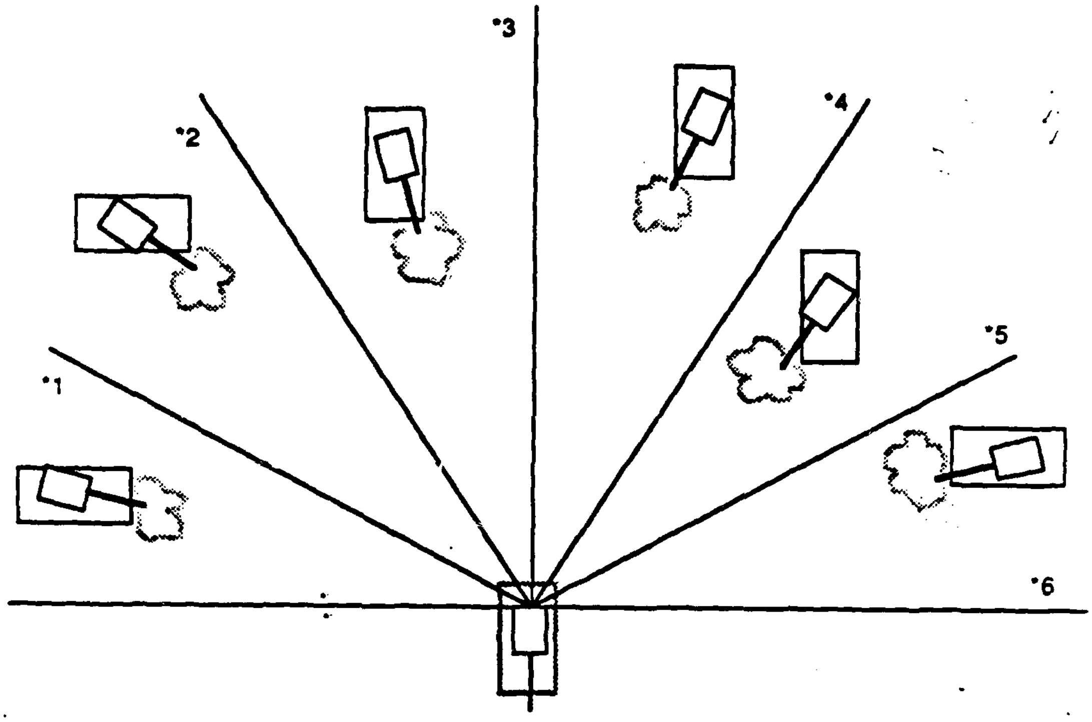
*3
Back Right 00-30° damages
90% kill - blast doors closed
100% kill - blast doors open
10% turret gearbox

*4
Back Left 00-30° damages
90% kill blast door closed
100% kill blast door open
10% turret gearbox

*5
Back Left 30-60° damages
60% kill blast door closed
100% kill blast door open
10% turret gearbox

*6
Back Left 60-90° damages
40% kill blast doors closed
100% kill blast doors open

Figure 10-11: Direct Fire Damage: HEAT hit - Turret Rear



SABOT Hits to Rear of M1 Turret

*1
Back Right 60-90° damages
40% kill - blast doors closed
100% kill - blast doors open
10% turret gearbox

*2
Back Right 30-60° damages
60% kill - blast doors closed
100 % kill - blast doors open
20% turret gearbox

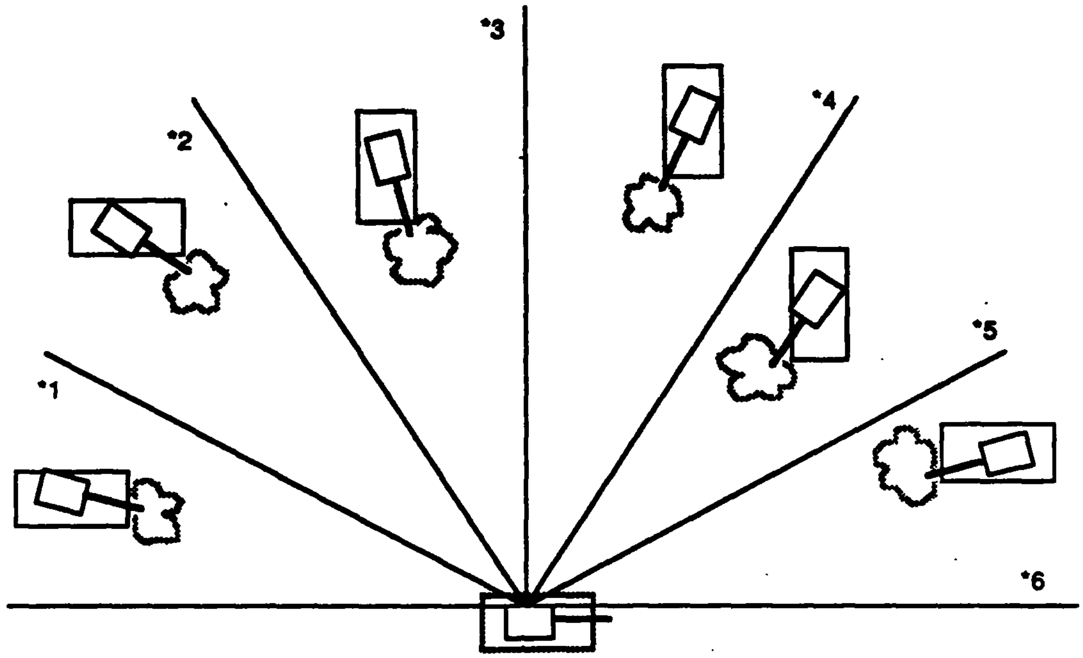
*3
Back Right 00-30° damages
90% kill - blast doors closed
100% kill - blast doors open
10% turret gearbox

*4
Back Left 00-30° damages
90% kill blast door closed
100% kill blast door open
10% turret gearbox

*5
Back Left 30-60° damages
60% kill blast door closed
100% kill blast door open
20% turret gearbox

*6
Back Left 60-90° damages
40% kill blast doors closed
100% kill blast doors open
10% turret gearbox

Figure 10-12: Direct Fire Damage: APDS hit - Turret Rear



*1
Left Back 60-90° damages
30% kill - blast doors closed
100% kill - blast doors open
10% turret gearbox

*2
Left Back 30-60° damages
40% kill - blast doors closed
100 % kill - blast doors open
20% gun mount / turret interface
20% turret gearbox

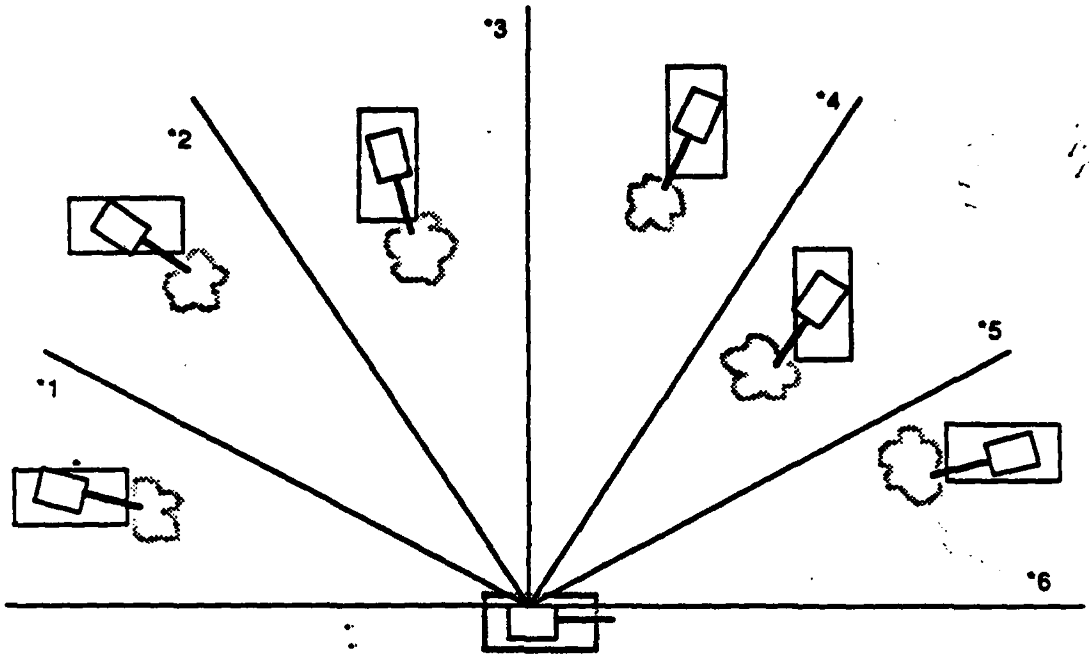
*3
Left Back 00-30° damages
50% kill - blast doors closed
100% kill - blast doors open
30% gun mount / turret interface
50% turret gearbox

*4
Left Front 00-30° damages
50% kill blast door closed
100% kill blast door open
30% gun mount / turret interface
50% turret gearbox

*5
Left Front 30-60° damages
40% kill blast door closed
100% kill blast door open
20% gun mount / turret interface
20% turret gearbox

*6
Left Front 60-90° damages
30% kill blast doors closed
100% kill blast doors open
10% turret gearbox

Figure 10-13: Direct Fire Damage: HEAT hit - Turret Left Side



SABOT Hits to Left Side
of M1 Turret

*1
Left Back 60-90° damages
30% kill - blast doors closed
100% kill - blast doors open
10% gun mount / turret interface
10% turret gearbox

*2
Left Back 30-60° damages
40% kill - blast doors closed
100 % kill - blast doors open
20% gun mount / turret interface
10% turret gearbox

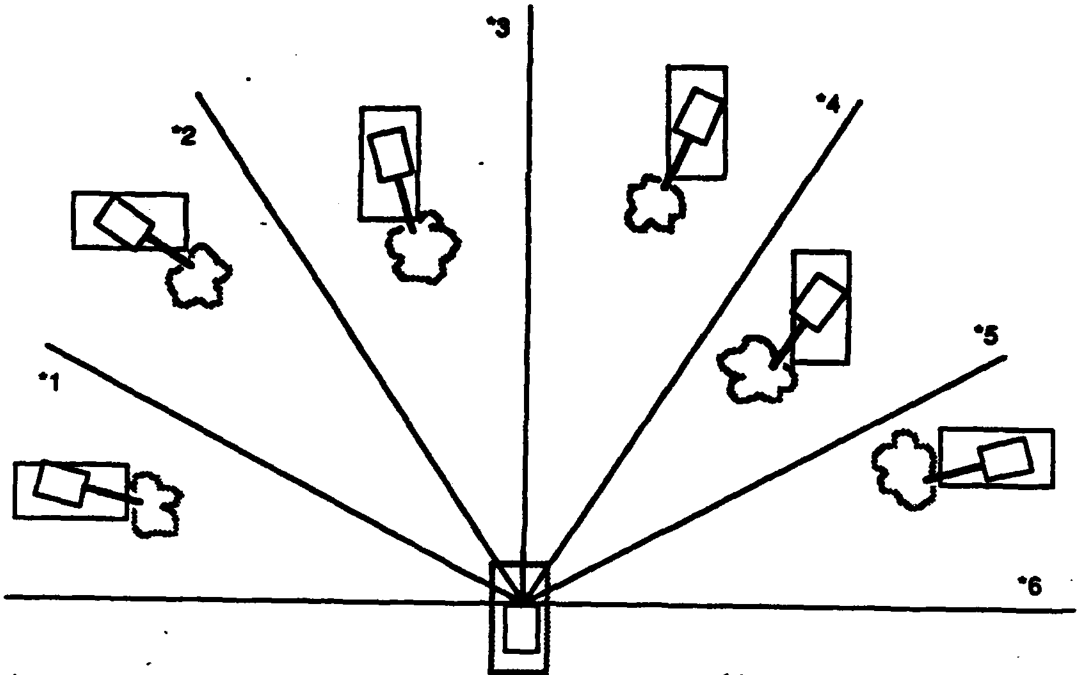
*3
Left Back 00-30° damages
50% kill - blast doors closed
100% kill - blast doors open
30% gun mount / turret interface
40% turret gearbox

*4
Left Front 00-30° damages
50% kill blast door closed
100% kill blast door open
30% gun mount / turret interface
40% turret gearbox

*5
Left Front 30-60° damages
40% kill blast door closed
100% kill blast door open
20% gun mount / turret interface
10% turret gearbox

*6
Left Front 60-90° damages
30% kill blast doors closed
100% kill blast doors open
10% gun mount / turret interface
10% turret gearbox

Figure 10-14: Direct Fire Damage: APDS hit - Turret Left Side



HEAT Hits to Bow of M1 Turret

*1
Front Left 60-90° damages
30% kill - blast doors closed
100% kill - blast doors open
10% gun mount / turret interface
20% turret gearbox

*2
Front Left 30-60° damages
35% kill - blast doors closed
100 % kill - blast doors open
20% LRF
20% GPS
10% CMDR GPS ext.
30% gun mount / turret interface
50% turret gearbox

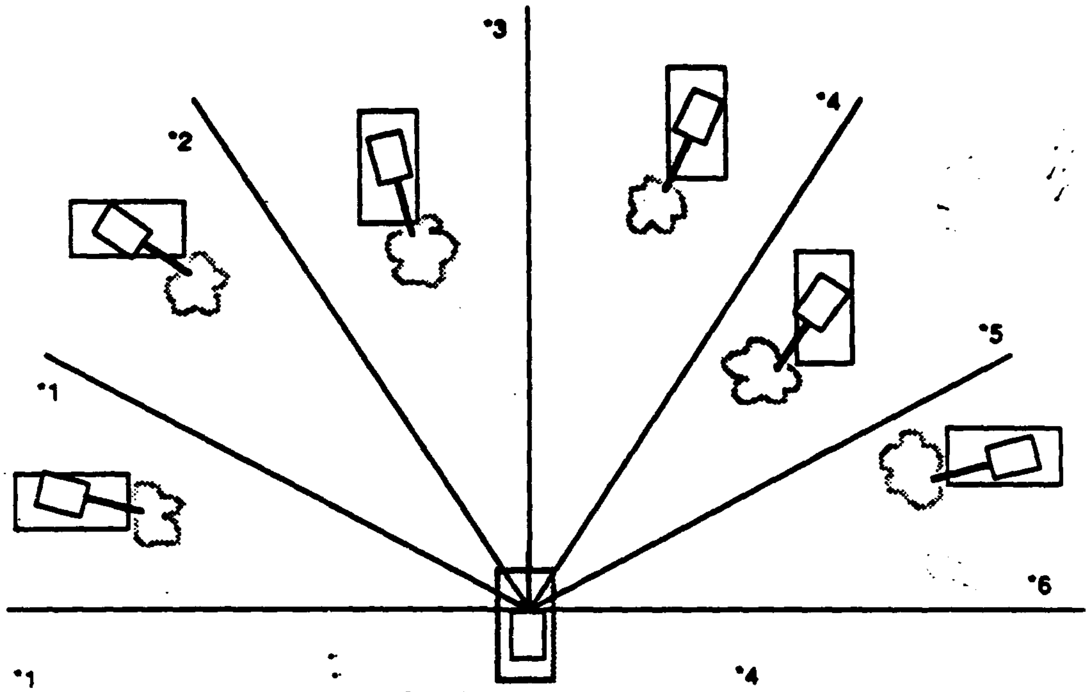
*3
Front Left 00-30° damages
50% kill - blast doors closed
100% kill - blast doors open
20% LRF
20% GPS
10% CMDR GPS ext
50% gun mount / turret interface

*4
Front Right 00-30° damages
50% kill blast door closed
100% kill blast door open
20% LRF
20% GPS
10% CMDR GPS ext
50% gun mount / turret interface

*5
Front Right 30-60° damages
35% kill blast door closed
100% kill blast door open
20% LRF
20% GPS
10% CMDR GPS ext
30% gun mount / turret interface
50% turret gearbox

*6
Front Right 60-90° damages
30% kill blast doors closed
100% kill blast doors open
10% gun mount / turret interface
20% turret gearbox

Figure 10-15: Direct Fire Damage: HEAT hit - Turret Front



SABOT Hits to Bow of M1 Turret

*1
Front Left 60-90° damages
30% kill - blast doors closed
100% kill - blast doors open
20% gun mount / turret interface
10% turret gearbox

*2
Front Left 30-60° damages
35% kill - blast doors closed
100 % kill - blast doors open
20% LRF
20% GPS
10% CMDR GPS ext.
40% gun mount / turret interface
40% turret gearbox

*3
Front Left 00-30° damages
50% kill - blast doors closed
100% kill - blast doors open
30% LRF
30% GPS
10% CMDR GPS ext.
60% gun mount / turret interface
20% turret gearbox

*4
Front Right 00-30° damages
50% kill blast door closed
100% kill blast door open
30% LRF
30% GPS
10% CMDR GPS ext
60% gun mount / turret interface
20% turret gearbox

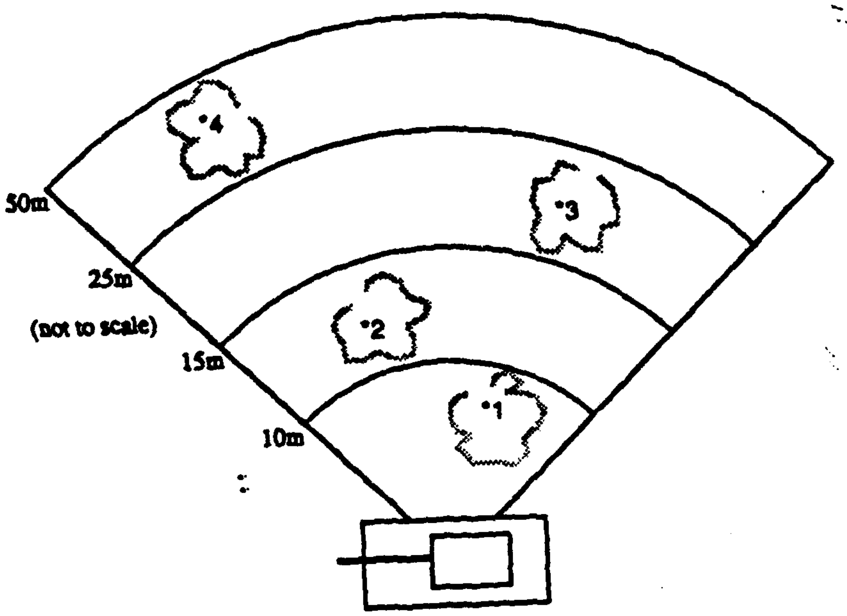
*5
Front Right 30-60° damages
35% kill blast door closed
100% kill blast door open
20% LRF
20% GPS
10% CMDR GPS ext
40% gun mount / turret interface
40% turret gearbox

*6
Front Right 60-90° damages
30% kill blast doors closed
100% kill blast doors open
20% gun mount / turret interface
10% turret gearbox

Figure 10-16: Direct Fire Damage: APDS hit - Turret Front

10.1.2 Indirect Fire

The following tables indicate the indirect fire damage probabilities of the M1 as implemented for the SIMNET simulator. The variable parameters that determine the indirect fire damage probabilities are direction to burst, and range to burst for PD and VT fuzes.



Indirect Fire at Right Side of M1 Hull

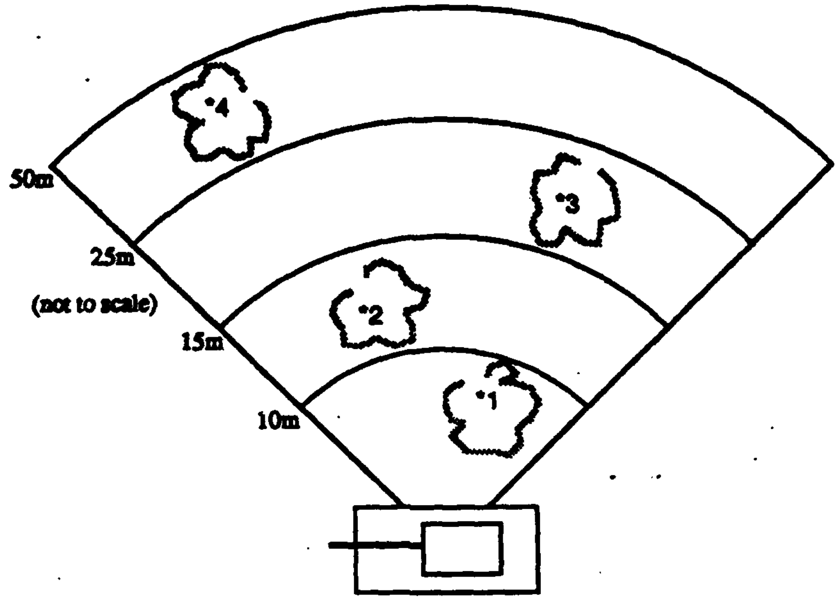
*1
Right Direct Hit (0-10 M) pd Fuze:
100% catastrophic kill

*2
Right Near Miss (10-15 M) pd Fuze:
80% commander vision block loss
50% driver vision loss
50% right track loss

*3
Right Far Miss (15-25 M) pd Fuze:
40% commander vision block loss
20% driver vision block loss
30% right track loss

*4
Right Miss (25-50 M) pd Fuze:
10% commander vision block loss
10% driver vision block loss

Figure 10-17: Indirect Fire Damage: PD fuze - Right Side



Indirect Fire at Right Side of M1 Hull

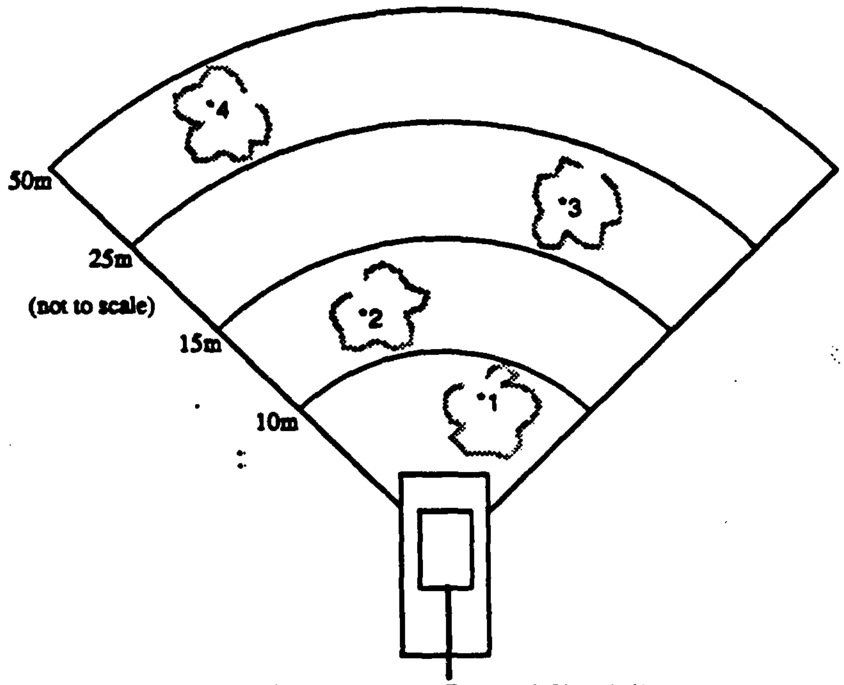
*1
Right Direct Hit (0-10 M) vt Fuze:
100% catastrophic kill

*2
Right Near Miss (10-15 M) vt Fuze:
80% commander vision block loss
50% driver vision loss
70% antenna loss
40% GPS loss

*3
Right Far Miss (15-25 M) vt Fuze:
40% commander vision block loss
20% driver vision block loss
20% antenna loss
20% GPS loss

*4
Right Miss (25-50 M) vt Fuze:
10% commander vision block loss
10% driver vision block loss
10% antenna loss
10% GPS loss

Figure 10-18: Indirect Fire Damage: VT fuze - Right Side



Indirect Fire at Rear of M1 Hull

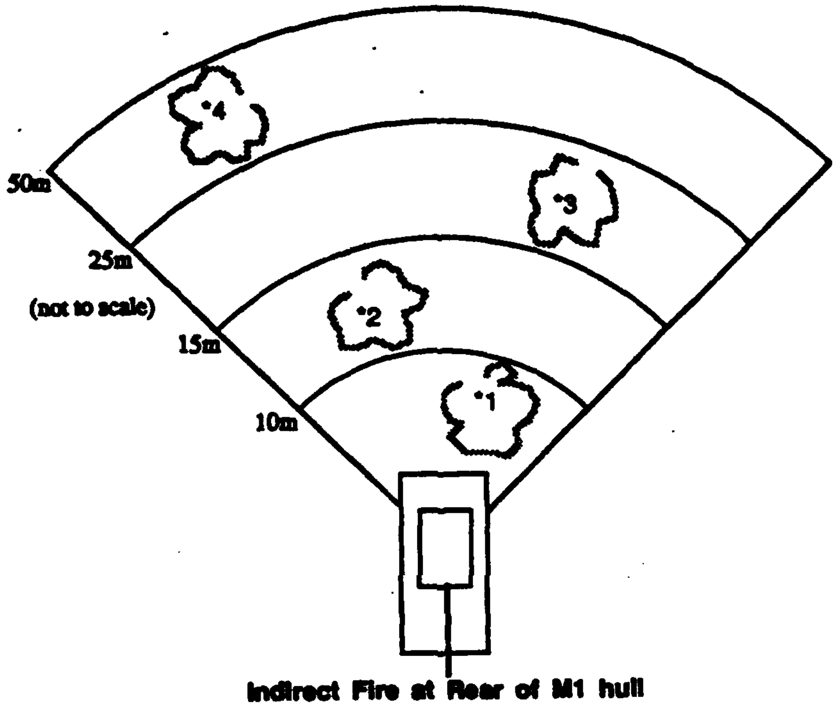
*1
Rear Direct Hit (0-10 M) pd Fuze:
100% catastrophic kill

*2
Rear Near Miss (10-15 M) pd Fuze:
70% loader vision block loss
70% commander vision block loss
30% left track loss
30% right track loss

*3
Rear Far Miss (15-25 M) pd Fuze:
30% loader vision block loss
30% commander vision block loss
20% left track loss
20% right track loss

*4
Rear Miss (25-50 M) pd Fuze:
10% loader vision block loss
10% commander vision block loss

Figure 10-19: Indirect Fire Damage: PD fuze - Rear



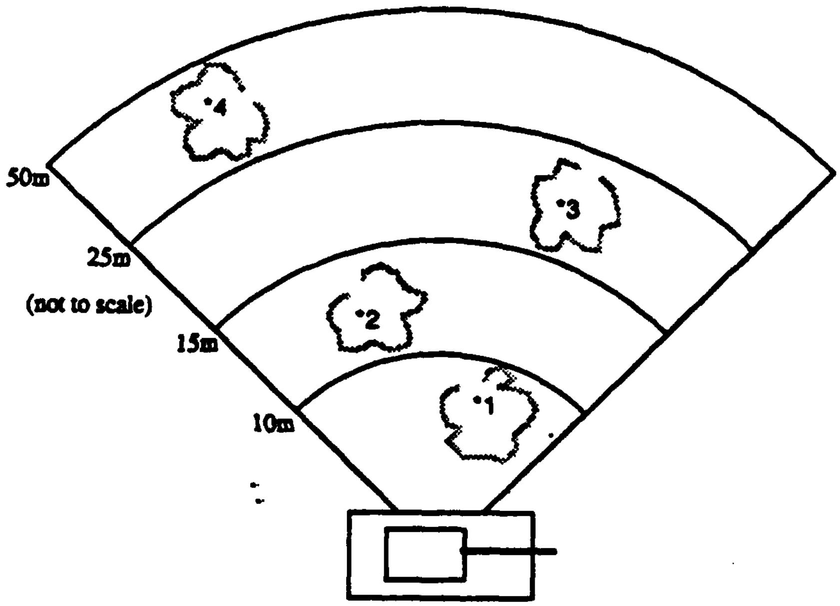
*1
Rear Direct Hit (0-10 M) vt Fuze:
100% catastrophic kill

*2
Rear Near Miss (10-15 M) vt Fuze:
70% loader vision block loss
70% commander vision block loss
70% chance antenna loss
40% chance GPS loss

*3
Rear Far Miss (15-25 M) vt Fuze:
30% loader vision block loss
30% commander vision block loss
50% antenna loss
20% GPS loss

*4
Rear Miss (25-50 M) vt Fuze:
10% loader vision block loss
10% commander vision block loss
10% antenna loss
10% GPS loss

Figure 10-20: Indirect Fire Damage: VT fuze - Rear



Indirect Fire at Left Side of M1 Hull

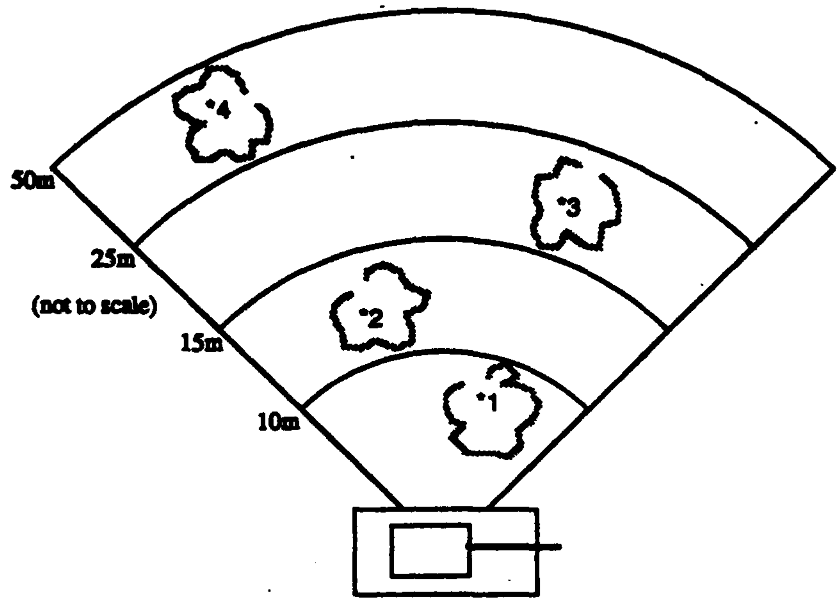
*1
Left Direct Hit (0-10 M) pd Fuze:
100% catastrophic kill

*2
Left Near Miss (10-15 M) pd Fuze:
80% loader vision block loss
50% driver vision loss
50% left track loss

*3
Left Far Miss (15-25 M) pd Fuze:
40% loader vision block loss
20% driver vision block loss
30% left track loss

*4
Left Miss (25-50 M) pd Fuze:
10% loader vision block loss
10% driver vision block loss

Figure 10-21: Indirect Fire Damage: PD fuze - Left Side



Indirect Fire at Left Side of M1 Hull

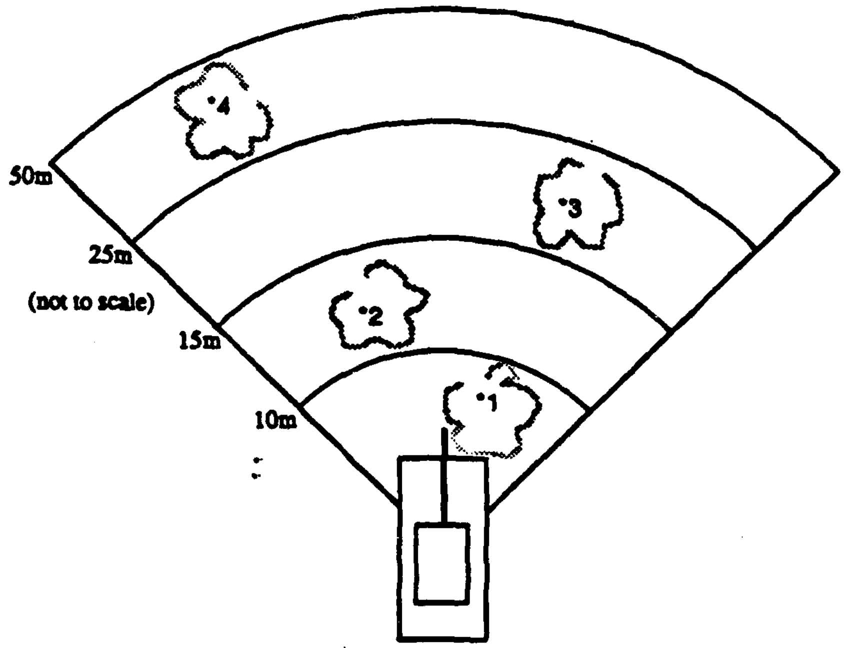
*1
Left Direct Hit (0-10 M) vt Fuze:
100% catastrophic kill

*2
Left Near Miss (10-15 M) vt Fuze:
80% loader vision block loss
50% driver vision loss
70% antenna loss
40% GPS loss

*3
Left Far Miss (15-25 M) vt Fuze:
40% loader vision block loss
20% driver vision block loss
50% antenna loss
20% GPS loss

*4
Left Miss (25-50 M) vt Fuze:
10% loader vision block loss
10% driver vision block loss
10% antenna loss
10% GPS loss

Figure 10-22: Indirect Fire Damage: VT fuze - Left Side



Indirect Fire at Bow of M1 Hull

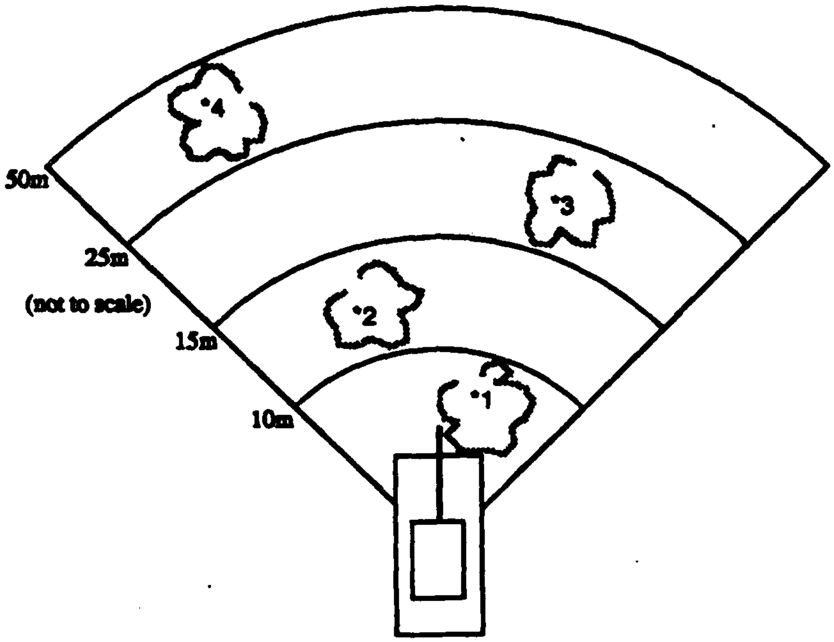
*1
Front Direct Hit (0-10 M) pd Fuze:
100% catastrophic kill

*2
Front Near Miss (10-15 M) pd Fuze:
40% loader vision block loss
40% commander vision block loss
60% driver vision block loss
30% left track loss
30% right track loss

*3
Front Far Miss (15-25 M) pd Fuze:
15% commander vision block loss
30% driver vision block loss
20% left track loss
20% right track loss

*4
Front Miss (25-50 M) pd Fuze:
10% loader vision block loss
10% commander vision block loss
10% driver vision block loss

Figure 10-23: Indirect Fire Damage: PD fuze - Front



Indirect Fire at Bow of M1 Hull

*1
Front Direct Hit (0-10 M) vt Fuze:
100% kill - blast doors closed

*2
Front Near Miss (10-15 M) vt Fuze:
40% loader vision block loss
40% commander vision block loss
60% driver vision block loss
70% antenna loss
40% GPS loss

*3
Front Far Miss (15-25 M) vt Fuze:
15% loader vision block loss
15% commander vision block loss
30% driver vision block loss
50% antenna loss
20% GPS loss

*4
Front Miss (25-50 M) vt Fuze:
10% loader vision block loss
10% commander vision block loss
10% driver vision loss
10% antenna loss
10% GPS loss

Figure 10-24: Indirect Fire Damage: VT fuze - Front

10.2 Stochastic Failures

For purposes of failure generation, the vehicle is divided into thirteen subsystems, six of which can encounter stochastic failures. The thirteen systems are:

- Mobility Electrical*
- Suspension
- Track
- Engine*
- Transmission*
- Final Drive
- Mobility Other*
- Fire Control*
- Turret Drive & Stabilization*
- Gun Mount
- Government Furnished Equipment (GFE)
- Non-mobility Electrical
- Non-mobility Other

* systems which are subject to stochastic failures

Each subsystem has associated with it a Mean Number of Operations Between Failures (MNOBF). These data were taken directly from Appendix I of LSS Data Package. Each subsystem has a set of specific failures associated with it, and each failure has a given probability of occurring if the subsystem fails. (See figure 10-25). The probability of any failure occurring when the subsystem fails was derived from the Table of Maintenance Demands during Operational Testing (Appendix I of LSS Data Package).

When a tank is initialized it receives a Maintenance Status Parameter (MSP) from the network. The MSP range is 1 - 5, where 1 represents a new tank, and 5 represents a tank which is 5 years old or has travelled 5000+ miles (see figure 10-26). The initial setting of the MNOBF is decreased for each subsystem when the MSP value is greater than one. This decrease allows the initial stochastic failure of a subsystem to occur sooner.

The stochastic failure procedure is run every time the vehicle has travelled a mile. For each subsystem, there is a certain percentage chance that the subsystem will fail. This chance is based on the MNOBF value for the subsystem. If a random number between 0 and 100 is less than this chance, then the subsystem fails. Once a subsystem has failed, then it must be determined which of the specific failures of that subsystem has failed. A random number is used to pick from these specific failures, based on the probability of each specific failure, and a routine is called to notify the appropriate module(s) affected by the failure.

Subsystem: Mobility Electrical		MNOBF: 2400 miles
alternator	.32	
pilot relay	.34	
starter	.34	
Subsystem: Engine		MNOBF: 3400 miles
engine oil filter clogged	.33	
engine oil low	.33	
major component	.34	
Subsystem: Transmission		MNOBF: 8000 miles
transmission oil low	.33	
trans oil filter clogged	.33	
forward/reverse assembly	.34	
Subsystem: Mobility Other		MNOBF: 1200 miles
fuel pump	.32	
fuel filter clogged	.32	
service brake	.18	
parking brake	.18	
Subsystem: Fire Control		MNOBF: 3000 miles
LRF receiver/transmitter	.16	
gunner GPS	.42	
commander GPS extension	.42	
Subsystem: Turret Drive & Stabilization		MNOBF: 1500 miles
gun stabilization	.26	
gun elevation	.12	
turret traverse	.12	
gunner handles	.25	
commander handles	.25	

Figure 10-25: Stochastic Failure Table

Code	Years/Miles	Failure Expectations
1	New/0-100 miles	Every 310 miles or 9 hours of operation
2	1 Year/1000 miles	Every 250 miles or 7 hours of operation
3	2 Years/2000 miles	Every 200 miles or 5 hours of operation
4	3 Years/3500 miles	Every 100 miles or 2.5 hours of operation
5	5 Years/5000+ miles	Every 40 miles or 2 hours of operation

Figure 10-26: Maintenence Status Parameter Codes

10.3 Deterministic Failures

Deterministic failures consist of failures due to resource depletion, throwing a track, damage resulting from a collision, and the failures that are a consequence of ignoring stochastic failure warnings.

There are several depletable resources in SIMNET with which the vehicle must be adequately supplied in order to function normally. These include hydraulic pressure, battery charge, fuel and ammunition. Hydraulic pressure can be replenished simply by turning on the engine or waiting for the auxiliary hydraulic pump to recharge the system. A low battery can be recharged by running the engine for a period of time or, if so low as to prevent starter operation, by waiting for a repair truck dispatched from the MCC Admin/Log console. Fuel and ammunition must be replenished from resupply vehicles also dispatched from the MCC Admin/Log console. For more information on ammunition and fuel management, see Chapter 8.

Other deterministic failures include:

1. If the engine is operated for 15 miles or 30 minutes with an clogged engine oil filter, a warning light will illuminate indicating low engine oil pressure. If the driver does not immediately stop and shut down the engine, damage to the engine will result, culminating in a major engine failure.
2. If the engine is operated for 15 miles or 30 minutes with an engine oil low condition, a warning light will illuminate indicating high engine oil temperature. If the driver does not immediately stop and shut down the engine, damage to the engine will result, culminating in a major engine failure.
3. If the vehicle is operated for 15 miles or 30 minutes with a clogged transmission oil filter, a warning light will illuminate indicating low transmission oil pressure. If the driver does not immediately stop and shut down, transmission failure will result.
4. If the vehicle is operated for 15 miles or 30 minutes with a transmission oil condition, a warning light will illuminate indicating high transmission oil temperature. If the driver does not immediately stop and shut down, transmission failure will result.
5. If the vehicle is driven across a slope greater than 25 degrees or if the driver attempts too sharp a turn at high speed in soft soil, a track will throw, effectively immobilizing the vehicle. However, the vehicle will still be able to move slowly (about 2 mph) so that the crew will be able to get the vehicle off the slope and commence a track repair.
6. If the vehicle is involved in a collision with a solid object (another tank, large tree or a building) at or above five miles per hour in the direction of the gun tube, the gun mount-turret interface will be damaged. This will result in a failure in which the gun cannot either be elevated or fired.

11. Repair Simulation

Repairs in SIMNET are classified in two categories: A) Self-repairs which represent those repairs that the crew can perform on their own without assistance, and B) Repairs via the MCC Maintenance console in which the crew must coordinate with the dispatcher to arrange a rendezvous with a repair maintenance vehicle. Both classes of repairs are timed using MTTR (mean time to repair) data.

11.1 Self-Repairs

Self-repairs commence automatically upon occurrence of those damages to the vehicle that a crew could repair themselves: Remounting thrown tracks, replacing damaged vision blocks, and replacing damaged radio antennas. The repair times for these failures are as follows:

Repairing thrown tracks:	30 minutes
Replacing damaged vision block:	10 minutes
Replacing damaged radio antenna:	10 minutes

These repairs are timed internally by the simulator and do not require any action to be taken by the crew. These timers start running as soon as the failure occurs and will automatically repair the damaged system when the repair time has elapsed.

11.2 Repairs via MCC Maintenance

Since every M1 subsystem can fail, it is necessary to simulate repairs from the MCC maintenance teams. The MCC can dispatch a maintenance team vehicle to a specified location. If a disabled M1 tank is close enough to a maintenance vehicle and satisfies a few other conditions, the maintenance team may begin repairs.

When the commander of the disabled M1 tank needs repairs, he radios to the maintenance officer at the MCC Admin/Log Center for help. The maintenance officer then dispatches the nearest available maintenance team vehicle to the disabled tank. In an amount of time proportional to the distance travelled, the maintenance team arrives. If the tank is alive and stationary, and there is a maintenance team within 20 meters, the tank requests service from the maintenance team. The maintenance team then repairs the broken subsystems, one repair at a time,

at a realistic rate, until the service request conditions are not satisfied. The service repair times range from the one-minute repair time for a clogged fuel filter to the four-hour repair time for a broken gun mount interface. While the repair is taking place, if the M1 tank or the maintenance team drives away or if either vehicle is destroyed, the repair will abort. As each repair is completed, the subsystem data structures are updated to note the repair.

See figure 11-1 for an illustration of the repair finite state machine.

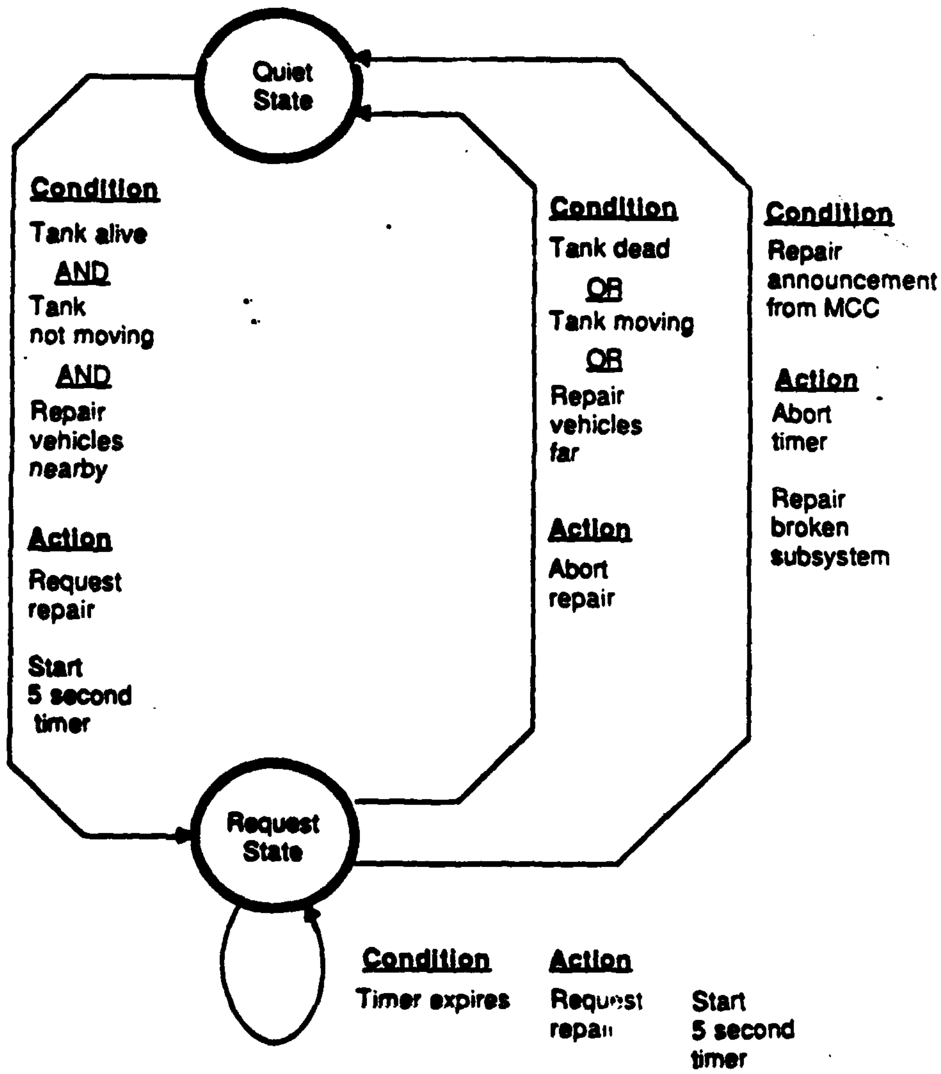


Figure 11-1: Repair Finite State Machine

The two repair states are QUIET and REQUEST, and the finite state machine operates as follows.

QUIET STATE:
If the following conditions are ALL true:
1. The tank is alive.
2. The tank is not moving.
3. There is a repair vehicle within a distance of 20 meters.
Then
1. The simulator sends a service request packet to the MCC.
2. The simulator starts a 5 second timer.
3. The simulator enters the REQUEST STATE.

REQUEST STATE:
If AT LEAST ONE of the following conditions is true:
1. The tank is dead.
2. The tank is moving.
3. There are no repair vehicles within a distance of 20 meters.
Then
1. The repair is aborted.
2. The simulator enters the QUIET STATE.

If the 5 second timer expires:
Then
1. The simulator sends another service request packet to the MCC.
2. The simulator starts a 5 second timer.
3. The simulator stays in the REQUEST STATE.

If the MCC announces that a successful repair has taken place on a broken subsystem of the tank:
Then
1. The simulator repairs the broken subsystem.
2. The 5 second timer is aborted.
3. The simulator enters the QUIET STATE.

By means of this finite state machine protocol and a similar one on the MCC host, the simulator and the MCC cooperate to repair disabled combat vehicles.

Notes

[1]"Characteristics and Description. M1 Abrams Tank", General Dynamics Land Systems Division, August 1982 p.6

[2]Ibid.. p.12

[3]Ibid.. p.13

[4]Ibid.. p.16

[5]Ibid.. p.16

[6]Ibid.. p.21

[7]Ibid.. p.72